

**RANCANG BANGUN SISTEM INFORMASI PENGENDALIAN
PERSEDIAAN BAHAN BAKU DENGAN MENGGUNAKAN
PHP 5.4.7 DAN MYSQL 5.5.27 PADA PT COCA COLA
AMATIL INDONESIA**

TUGAS AKHIR

Untuk Memenuhi Sebagian Syarat Penyelesaian Program Studi Diploma Empat
(D-IV) Program Studi Sistem Informasi Pada Sekolah Tinggi Manajemen Industri

**OLEH
MOHAMMAD ALAM MUNTAHA
1310041**



**SEKOLAH TINGGI MANAJEMEN INDUSTRI
KEMENTERIAN PERINDUSTRIAN RI
JAKARTA
2016**

SEKOLAH TINGGI MANAJEMEN INDUSTRI
KEMENTERIAN PERINDUSTRIAN R.I.

TANDA PERSETUJUAN DOSEN PEMBIMBING

Judul Tugas Akhir : *Rancang Bangun Sistem Informasi Pengendalian Persediaan Bahan Baku Dengan Menggunakan Php 5.4.7 Dan MySQL 5.5.27 Pada PT Coca Cola Amatil Indonesia*

Disusun Oleh

Nama : Mohamad Alam Muntaha
Nim : 1310041
Program Studi : Sistem Informasi

Telah diperiksa dan disetujui untuk diajukan dan dipertahankan dalam sidang Tugas Akhir Sekolah Tinggi Manajemen Industri

Jakarta, Februari 2016

Dosen Pembimbing

Drs.Jacob Saragih,MM

NIP. 195404281986031002

**SEKOLAH TINGGI MANAJEMEN INDUSTRI
KEMENTERIAN PERINDUSTRIAN R.I.**

TANDA PERSETUJUAN ASISTEN DOSEN PEMBIMBING

Judul Tugas Akhir : *Rancang Bangun Sistem Informasi Pengendalian Persediaan Bahan Baku Dengan Menggunakan Php 5.4.7 Dan MySQL 5.5.27 Pada PT Coca Cola Amatil Indonesia*

Disusun Oleh

Nama : Mohamad Alam Muntaha
Nim : 1310041
Program Studi : Sistem Informasi

Telah diperiksa dan disetujui untuk diajukan dan dipertahankan dalam sidang Tugas Akhir Sekolah Tinggi Manajemen Industri.

Jakarta, Februari 2016
Assisten Dosen Pembimbing

Fifi L.Hadianastuti, S.Kom, M.Kes
NIP. 197310162005022001

SEKOLAH TINGGI MANAJEMEN INDUSTRI
KEMENTERIAN PERINDUSTRIAN R.I.

LEMBAR BIMBINGAN TUGAS AKHIR

Nama : Mohamad Alam Muntaha
Nim : 1310041
Judul Tugas Akhir : *Rancang Bangun Sistem Informasi Pengendalian Persediaan Bahan Baku Dengan Menggunakan Php 5.4.7 Dan MySQL 5.5.27 Pada PT Coca Cola Amatil Indonesia*
Dosen Pembimbing : Drs.Jacob Saragih,MM

Tanggal	Keterangan	Paraf
05 Mei 2015	Diskusi Awal	
19 Mei 2015	Konsultasi Proposal	
30 Mei 2015	Konsultasi Bab I dan Bab II	
22 Juni 2015	Konsultasi Bab II dan Bab III	
07 Juli 2015	Konsultasi Bab III dan Bab IV	
01 September 2015	Konsultasi Bab V dan Bab VI	
03 Oktober 2015	Konsultasi Program	
03 November 2015	Koreksi Ulang Bab I – Bab VI	
29 Januari 2016	Koreksi Ulang Program	
16 Februari 2016	Pra Presentasi	

Jakarta, Februari 2016

Mengetahui,
Dosen Pembimbing

Drs.Jacob Saragih,MM

NIP. 195404281986031002

SEKOLAH TINGGI MANAJEMEN INDUSTRI
KEMENTERIAN PERINDUSTRIAN R.I.

LEMBAR BIMBINGAN TUGAS AKHIR

Nama : Mohamad Alam Muntaha
Nim : 1310041
Judul Tugas Akhir : *Rancang Bangun Sistem Informasi Pengendalian Persediaan
Bahan Baku Dengan Menggunakan Php 5.4.7 Dan MySQL 5.5.27
Pada PT Coca Cola Amatil Indonesia*
Asisten Dosen Pembimbing : Fifi L.Hadianastuti, S.Kom, M,Kes

Tanggal	Keterangan	Paraf
18 Mei 2015	Diskusi Awal	
28 Mei 2015	Proposal TA	
18 Juni 2015	Bahas Bab I	
06 Juli 2015	Bahas Bab II	
31 Agustus 2015	Bahas Bab III	
29 September 2015	Bahas Bab IV	
30 Oktober 2015	Bahas Bab V	
26 Januari 2016	Bahas Bab VI	
11 Februari 2016	Bahas Bab I – Bab VI	

Jakarta, Februari 2016

Mengetahui,

Asisten Dosen Pembimbing

Fifi L.Hadianastuti, S.Kom, M,Kes

NIP. 197310162005022001

LEMBAR PERNYATAAN KEASLIAN

Saya yang bertanda tangan di bawah ini:

Nama : Mohamad Alam Muntaha

Nim : 1310041

Berstatus sebagai mahasiswa Program Studi Sistem Informasi di Sekolah Tinggi Manajemen Industri Kementerian Perindustrian RI. Dengan ini menyatakan bahwa hasil karya Tugas Akhir yang saya buat dengan judul :

“RANCANG BANGUN SISTEM INFORMASI PENGENDALIAN PERSEDIAAN BAHAN BAKU DENGAN MENGGUNAKAN PHP 5.4.7 DAN MYSQL 5.5.27 PADA PT COCA COLA AMATIL INDONESIA”

- **Dibuat** dan diselesaikan sendiri, dengan menggunakan literatur hasil kuliah, *survey* lapangan, dibantu oleh dosen pembimbing maupun asisten dosen pembimbing, serta buku-buku maupun jurnal-jurnal ilmiah yang menjadi bahan acuan yang tertera dalam referensi karya Tugas Akhir ini.
- **Bukan** merupakan hasil duplikasi karya tulis yang sudah dipublikasikan atau yang pernah dipakai sebelumnya untuk mendapatkan gelar sarjana di Universitas/Perguruan Tinggi lain, kecuali yang telah disebutkan sumbernya dan dicantumkan pada referensi karya Tugas Akhir ini.
- **Bukan** merupakan karya tulis hasil terjemahan dari kumpulan buku atau jurnal acuan yang tertera dalam referensi karya Tugas Akhir ini.

Jika terbukti tidak memenuhi apa yang telah saya nyatakan diatas, maka saya bersedia menerima sanksi atas apa yang telah saya lakukan sesuai peraturan yang berlaku.

Jakarta, September 2016

Mohamad Alam Muntaha

ABSTRAK

PT Coca Cola Amatil Indonesia adalah perusahaan manufaktur yang bergerak dalam produksi minuman ringan, dengan kualitas yang sudah mendunia. Dalam proses pengendalian persediaan bahan baku bagian produksi dalam membuat *material list* bahan baku masih menggunakan *microsoft excel*, Bagian gudang dalam mengecek ketersediaan stok bahan baku hanya melihat jumlah fisik yang ada digudang, sehingga dibutuhkan waktu yang lama dalam melakukan kegiatan tersebut. Bagian *Inventory Material* yang mempunyai wewenang dalam membuat laporan *purchase requisition* dan *purchase order* masih menemukan beberapa kekurangan dalam menggunakan *microsoft excel*, sehingga dalam membuat pengolahan data dan laporan kurang terintegrasi. Untuk mengatasi permasalahan tersebut maka perlu dibuat sebuah sistem informasi yang dapat memudahkan pengendalian persediaan bahan baku pada PT Coca Cola Amatil Indonesia. Untuk membuat sistem tersebut maka diperlukan pengumpulan data dengan cara pengamatan langsung pada departemen *Inventory Material* di PT Coca Cola Amatil Indonesia. Data yang telah diperoleh selanjutnya akan diolah menjadi informasi yang diperlukan. Metodologi pengembangan sistem informasi yang digunakan pada penelitian ini adalah metodologi *evolutionary prototype*. Kemudian dalam merancang sistem informasi Pengendalian Persediaan bahan baku pemodelan sistem menggunakan *unified modelling language* (UML). Sistem informasi Pengendalian persediaan bahan baku dibangun dengan menggunakan PHP 5.4.7 sebagai bahasa pemrograman dan MYSQL 5.2.27 sebagai basis datanya. Sistem informasi Pengendalian Persediaan bahan baku ini akan memudahkan bagian *Inventory Material* dalam membuat laporan *purchase order* dan *purchase requisition* secara terintegrasi.

Kata kunci: Sistem Informasi, Pengendalian Persediaan Bahan Baku, *Evolutionary Prototype*, UML, PHP, *Microsoft Excel*.

KATA PENGANTAR

Bismillahirrohmanirrohim

Assalamu 'alaikum Warohmatullohi Wabarokatuh.

Alhamdulillah Robbilalamin, yang berarti segala puji bagi Allah SWT, Tuhannya Semesta Alam. Sungguhlah ucapan ini yang banyak diucapkan oleh penulis untuk mensyukuri apa yang telah Allah SWT berikan kepada penulis dan seluruh alam semesta beserta apa – apa yang ada di dalamnya. Kemudian Shalawat serta salam semoga tercurah bagi Nabi Muhammad SAW, seorang sosok suri tauladan bagi umat seluruh alam atas seluruh perilaku mulianya beliau yang patut dicontoh, ditiru dan di aplikasikan kepada setiap perilaku manusia di muka bumi, agar tercapainya manusia yang unggul, profesional, dan religius. Seperti apa yang Nabi Muhammad telah contohkan dan aplikasikan ke dalam kehidupannya sehari hari. Penyusunan laporan tugas akhir ini pada hakikatnya dengan judul **RANCANG BANGUN SISTEM INFORMASI PENGENDALIAN PERSEDIAAN BAHAN BAKU DENGAN MENGGUNAKAN PHP 5.4.7 dan MySQL 5.5.27 PADA PT COCA COLA AMATIL INDONESIA** sebagai laporan tugas akhir di PT COCA COLA AMATIL INDONESIA ini ditunjukan untuk memenuhi salah satu syarat akademis untuk menyelesaikan Program Studi D-IV Sistem Informasi di Sekolah Tinggi Manajemen Industri Kementerian Perindustrian Republik Indonesia.

Dalam penyusunan laporan tugas akhir ini penulis banyak menerima bantuan baik materi maupun moril. Untuk itu penulis mengucapkan terima kasih yang sebesar besarnya terutama kepada:

1. Allah SWT yang telah memberikan sangat banyak rahmat, kelancaran, kebarokahan, keamanan, serta kemudahan kepada penulis dalam mengerjakan laporan Tugas Akhir.
2. Orang tuaku Bapak Muchlis Wicaksono, Ibu Endang Winarni, saudara-saudaraku M Reza Pahlevi, dan Anissa Tri Anggraini yang tidak pernah

lelah mendo'akan dan memberikan dorongan moral serta dukungan finansial kepada penulis hingga selesainya penyusunan laporan Tugas Akhir ini.

3. Bapak Drs. Achmad Zawawi, MA, MM, selaku Ketua Sekolah Tinggi Manajemen Industri Kementerian Perindustrian RI beserta dengan jajarannya.
4. Bapak Dedy Trisanto, S.Kom, MMSI, selaku Ketua Program Studi Sistem Informasi Sekolah Tinggi Manajemen Industri Kementerian Perindustrian RI beserta dengan jajarannya.
5. Bapak Drs.Jacob Saragih,MM selaku dosen pembimbing yang dengan penuh kesabaran telah banyak meluangkan waktu dan pemikirannya untuk memberikan petunjuk, saran, arahan serta bimbingan yang sangat berharga kepada penulis selama proses penyusunan laporan tugas akhir ini.
6. Ibu Fifi L.Hadianastuti S.Kom, M.Kes selaku asisten dosen pembimbing yang telah memberikan arahan, bimbingan serta penjelasan dalam penyusunan tugas Akhir.
7. Bapak Asep Sunandar selaku Kepala Bagian *Inventory Material* dan seluruh staff yang dengan segala kesibukannya telah berkenan membimbing dan mengarahkan penulis selama melaksanakan penelitian di PT COCA COLA AMATIL INDONESIA.
8. Sahabat sahabatku Angkatan 2010 dan terutamanya kepada Andika Kusuma, Muhammad Aditya Suazi, Tri Andriano Ibrahim, Ahmad Muzakki Mubarak, Mamik Restianti, dan Faqih Danurliamto yang telah membantu penulis menyelesaikan laporan tugas akhir ini.
9. Teman spesial yang telah menyemangatiku Desti Anitaputri, S. Kep, Ners.
10. Para kawan-kawan Toko Bang Bob dan yang telah memberiku banyak kesempatan untuk terus memperkaya wawasan dan pengetahuan diri.
11. Kepada semua pihak yang tidak dapat penulis sebutkan satu persatu, yang telah memberikan dan menumbuhkan semangat dan motivasi serta do'a yang terus terucap, baik secara langsung ataupun tidak langsung.

Penulis menyadari sepenuhnya bahwa penyusunan tugas akhir ini masih banyak terdapat kekurangan dan sangat jauh dari kesempurnaan. Hal ini dikarenakan keterbatasan pengetahuan yang penulis miliki. Oleh karena itu dengan segala kerendahan hati, penulis sangat mengharapkan kritik dan saran yang bersifat membangun dari berbagai pihak guna memperbaiki serta menyempurnakan laporan tugas akhir ini.

Besar harapan penulis agar laporan tugas akhir ini dapat bermanfaat bagi para pembaca dan dapat menjadi sebuah pembelajaran bagi proses proses penelitian berikutnya

Alhamdulillah jaza kumullohu khoiro.

Wassalamu'alaikum Warahmatullahi Wabarakatuh.

Bekasi, Januari 2016

Penulis

DAFTAR ISI

	Halaman
Halaman Judul	
Lembar Persetujuan Dosen Pembimbing	
Lembar Bimbingan Tugas Akhir	
Lembar Pernyataan Keaslian	
Abstrak	
Kata Pengantar	
Daftar Isi	xi
Daftar Gambar	xv
Daftar Tabel	xvii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Pokok Permasalahan	2
1.3 Tujuan Penelitian	2
1.4 Batasan Masalah.....	3
1.5 Manfaat Tugas Akhir	3
1.6 Sistematika Penulisan	4
BAB II LANDASAN TEORI	6
2.1 Konsep Dasar Sistem	6
2.1.1 Jenis Sistem.....	10

2.2	Konsep Dasar Informasi.....	11
2.2.1	Kualitas Informasi.....	12
2.2.2	Siklus Informasi	12
2.3	Pengertian Persediaan	13
2.3.1	Jenis-jenis Persediaan	15
2.3.2	Faktor-Faktor Yang Mempengaruhi Persediaan	17
2.3.3	Peranan Perencanaan dan Pengendalian Persediaan	17
2.4	Pengembangan Sistem	19
2.4.1	Model Prototipe Secara Umum.....	21
2.4.2	<i>Evolutionary Prototype</i>	23
2.5	<i>Flowchart</i> (Bagan Alir).....	25
2.6	<i>Unified Modeling Language</i> (UML).....	27
2.6.1	<i>Use Case Diagram</i>	28
2.6.2	<i>Activity Diagram</i>	29
2.6.3	<i>Sequence Diagram</i>	30
2.6.4	<i>Class Diagram</i>	32
2.6.5	<i>Deployment Diagram</i>	32
2.7	Kamus Data	33
2.8	HIPO (<i>Hierarchy plus Input Proses Output</i>)	35
2.9	XAMPP Local Server	37
2.10	PHP	38
2.11	MySQL.....	39

BAB III	METODOLOGI PENELITIAN.....	43
3.1	__Jenis dan Sumber Data.....	43
3.2	Metode Pengumpulan Data.....	43
3.3	Metode Pengembangan Sistem	44
3.4	Kerangka Penelitian	45
BAB IV	PENGUMPULAN DAN PENGOLAHAN DATA	48
4.1	Pengamatan dan Pengumpulan Data.....	48
4.2	<u>Sejarah Perusahaan.....</u>	<u>48</u>
4.2.1	<u>Visi dan Misi PT. Coca Cola Amatil</u>	<u>49</u>
4.2.2	<u>Tujuan Perusahaan</u>	<u>50</u>
4.2.3	Struktur Organisasi	50
4.3	Prosedur Sistem Informasi yang sedang Berjalan.....	53
4.3.1	Dokumen Terkait Dengan Sistem Informasi Pengendalian Persediaan Bahan Baku yang Berjalan	55
4.4	Sistem Informasi Pengendalian Persediaan Bahan Baku Produksi dengan <i>Unified Modeling Laguage</i> (UML).....	58
BAB V	ANALISIS DAN PEMBAHASAN	66
5.1	Analisis Kebutuhan <i>User</i>	66
5.2	Analisis Sistem	68
5.2.1	<i>Flowmap</i> Diagram yang Diusulkan	68
5.2.2	<i>Use Case</i> Diagram yang Diusulkan	70
5.2.3	<i>Activity</i> Diagram yang Diusulkan	73
5.2.4	<i>Sequence</i> Diagram yang Diusulkan	80
5.2.5	<i>Class</i> Diagram yang Diusulkan.....	87
5.2.6	Kamus Data.....	89

5.2.7 Deployment Diagram yang Diusulkan.....	92
5.3 Perancangan Sistem	93
5.3.1 <i>Hierarchy plus Input-Proses-Output</i> (HIPO).....	94
5.3.2 <i>Flowchart</i> Aplikasi Pengendalian Bahan Baku	95
5.4 Perancangan Input.....	96
5.4 Spesifikasi Perangkat Lunak dang Perangkat Keras.....	103
BAB VI KESIMPULAN DAN SARAN	104
6.1 Kesimpulan	104
6.2 Saran.....	104
DAFTAR PUSTAKA	105
LAMPIRAN	L-1

DAFTAR GAMBAR

Gambar II.1	Keterkaitan Antar Komponen dan Karakteristik Suatu Sistem	9
Gambar II.2	Model Umum Suatu Sistem.....	9
Gambar II.3	Siklus Informasi.....	13
Gambar II.4	Ilustrasi Model Prototipe	22
Gambar II.5	Evolutionary Prototype Model	24
Gambar II.6	Klasifikasi Diagram UML	27
Gambar II.7	Contoh Visual Table Of Contents	36
Gambar II.8	Contoh <i>Overview Diagram</i>	37
Gambar II.9	Siklus PHP dalam <i>Web Server</i>	39
Gambar II.10	Skema <i>Prototype Server</i> MySQL	41
Gambar III.1	Kerangka Pemecahan Masalah.....	47
Gambar IV.1	Struktur Organisasi PT Coca Cola Amatil Cibitung <i>Plant</i>	51
Gambar IV.2	<i>Flowmap</i> Pengendalian Persediaan Bahan Baku	54
Gambar IV.3	Dokumen Material List	56
Gambar IV.4	Dokumen <i>Purchase Requisition</i>	57
Gambar IV.5	Dokumen <i>Purchase Order</i>	57
Gambar IV.6	<i>Use Case</i> Sistem Pengendalian Bahan Baku pada PT CCAI Cibitung Plant	59
Gambar V.1	<i>FlowMap Diagram</i> yang Diusulkan.....	69
Gambar V.2	<i>Use Case Diagram</i> yang Diusulan.....	70
Gambar V.3	<i>Activity Diagram</i> Proses <i>Login</i> yang Diusulkan	74
Gambar V.4	<i>Activity Diagram</i> Membuat Material List.....	75
Gambar V.5	<i>Activity Diagram</i> Mengecek Ketersediaan Stok Bahan Baku.....	76
Gambar V.6	<i>Activity Diagram</i> <i>Purchase Requisition</i>	77
Gambar V.7	<i>Activity Diagram</i> <i>Purchase Order</i>	78
Gambar V.8	<i>Activity Diagram</i> Barang Masuk yang Diusulkan	79
Gambar V.9	<i>Activity Diagram</i> Barang Keluar yang Diusulkan.....	80

Gambar V.10	<i>Sequence Diagram Login</i> yang Diusulkan.....	81
Gambar V.11	<i>Sequence Diagram</i> Membuat Material List	82
Gambar V.12	<i>Sequence Diagram</i> Mengecek Ketersediaan Bahan Baku	83
Gambar V.13	<i>Sequence Diagram</i> Membuat <i>Purchase Requisition</i>	84
Gambar V.14	<i>Sequence Diagram</i> Membuat <i>Purchase Order</i>	85
Gambar V.15	<i>Sequence Diagram</i> Barang Keluar yang Diusulkan.....	86
Gambar V.16	<i>Sequence Diagram</i> Barang Masuk yang Diusulkan.....	87
Gambar V.17	<i>Class Diagram</i> Aplikasi Pengendalian Bahan Baku.....	88
Gambar V.18	<i>Deployment Diagram</i> yang Diusulkan	93
Gambar V.19	HIPO Aplikasi Pengendalian Persediaan Bahan Baku.....	94
Gambar V.20	<i>Flowchart</i> Aplikasi Pengendalian Bahan Baku.....	95
Gambar V.21	Perancangan Form <i>Login</i>	96
Gambar V.22	Perancangan Data User	96
Gambar V.23	Perancangan Data Bahan Baku	97
Gambar V.24	Perancangan Data Supplier	97
Gambar V.25	Perancangan Barang Masuk	98
Gambar V.26	Perancangan Barang Keluar	99
Gambar V.27	Perancangan Material List.....	99
Gambar V.28	Perancangan Form <i>Purchase Requisition</i>	100
Gambar V.29	Perancangan Form <i>Purchase Order</i>	100
Gambar V.30	Perancangan Laporan Barang Masuk.....	101
Gambar V.31	Perancangan Laporan Barang Keluar.....	101
Gambar V.32	Perancangan Request Material	102
Gambar V.33	Perancangan Laporan Stok Bahan Baku	108

DAFTAR TABEL

Tabel II.1	Simbol-Simbol <i>Flowchart</i>	25
Tabel II.2	Simbol-simbol <i>Use Case Diagram</i>	28
Tabel II.3	Simbol-simbol <i>Activity Diagram</i>	30
Tabel II.4	Simbol-simbol <i>Sequence Diagram</i>	31
Tabel II.5	Simbol-simbol <i>Class Diagram</i>	32
Tabel II.6	Simbol-simbol <i>Deployment Diagram</i>	33
Tabel II.7	Tipe Data pada MySQL	39
Tabel IV.1	Definisi Aktor dalam Sistem Informasi Pengendalian Persediaan Bahan Baku	60
Tabel IV.2	Definisi <i>Use Case</i> dalam Sistem Informasi Pengendalian Persediaan Bahan Baku	60
Tabel IV.3	Skenario <i>Use Case</i> Membuat <i>Material List</i>	62
Tabel IV.4	Skenario <i>Use Case</i> Menerima <i>Material List</i>	62
Tabel IV.5	Skenario <i>Use Case</i> Mengecek Ketersediaan Stok Bahan Baku	62
Tabel IV.6	Skenario <i>Use Case</i> Men- <i>checklist Material List</i>	63
Tabel IV.7	Skenario <i>Use Case</i> Mengirim Bahan Baku	63
Tabel IV.8	Skenario <i>Use Case</i> Melakukan Proses Produksi	63
Tabel IV.9	Skenario <i>Use Case</i> Menentukan Jumlah Bahan Baku yang Dipesan...	64
Tabel IV.10	Definisi Aktor dan <i>Use Case</i> Membuat <i>Purchase Requisition</i>	64
Tabel IV.11	Definisi Aktor dan <i>Use Case</i> Membuat <i>Purchase Order</i>	64
Tabel IV.12	Definisi Aktor dan <i>Use Case</i> Memberi bahan baku yang dipesan	65
Tabel V.1	Identifikasi Kebutuhan <i>User</i>	66
Tabel V.2	Skenario <i>Use Case</i> <i>Login</i>	70
Tabel V.3	Skenario <i>Use Case</i> Membuat <i>Material List</i>	71
Tabel V.4	Skenario <i>Use Case</i> Mengecek Ketersediaan Bahan Baku	71
Tabel V.5	Skenario <i>Use Case</i> Mengolah Data Master	72
Tabel V.6	Skenario <i>Use Case</i> Membuat <i>Purchase Requisition</i>	72

Tabel V.7	Skenario <i>Use Case</i> Membuat <i>Purchase Order</i>	72
Tabel V.8	Skenario <i>Use Case</i> Membuat Laporan Barang Keluar	73
Tabel V.9	Skenario <i>Use Case</i> Membuat Laporan Barang Masuk	73
Tabel V.10	Spesifikasi Tabel <i>User</i>	89
Tabel V.11	Spesifikasi Tabel <i>Master</i> Bahan Baku	89
Tabel V.12	Spesifikasi Tabel <i>Master Supplier</i>	90
Tabel V.13	Spesifikasi Tabel Barang Masuk.....	90
Tabel V.14	Spesifikasi Tabel <i>Purchase Requisition</i>	90
Tabel V.15	Spesifikasi Tabel <i>Purchase Order</i>	91
Tabel V.16	Spesifikasi Tabel Barang Keluar.....	91
Tabel V.17	Spesifikasi Tabel <i>Material List</i>	91
Tabel V.18	Spesifikasi Tabel <i>Request Material</i>	92
Tabel V.19	Implementasi Perangkat Keras.....	103

BAB I

PENDAHULUAN

1.1. Latar Belakang

Perkembangan sistem informasi dewasa ini sudah berkembang dengan sangat pesat seiring dengan kemajuan teknologi saat ini, sistem informasi merupakan hal yang penting dalam sebuah perusahaan. Dengan adanya sistem informasi, perusahaan dapat menjamin kualitas informasi yang dapat disajikan dengan baik, cepat dan mudah dipahami.

Berkembangnya suatu industri tidak terlepas dari peran penting sistem persediaan bahan baku. Oleh karena itu persediaan bahan baku perlu dilakukan pengendalian agar proses produksi berjalan dengan baik. Berdasarkan hasil penelitian pada berbagai perusahaan industri manufaktur, diperoleh pernyataan bahwa pengendalian persediaan bahan baku merupakan biaya terbesar pada sebuah perusahaan industri manufaktur. Persaingan pasar yang semakin ketat menuntut perusahaan untuk bekerja lebih efektif dan efisien. Ditambah dengan kenyataan bahwa pertumbuhan pasar jauh lebih besar dibandingkan dengan kemampuan produksi total industri, sehingga diperlukan ketepatan waktu.

PT Coca Cola Amatil Indonesia merupakan salah satu perusahaan produsen penghasil minuman terbesar di Asia Tenggara. Perusahaan ini memproduksi kurang lebih dua belas produk yang terbagi dalam tiga belas line produksi. Dengan banyaknya produk yang dihasilkan perusahaan ini tentunya akan memerlukan bahan baku yang banyak pula.

Dalam proses pengendalian persediaan bahan baku bagian produksi dalam membuat *material list* bahan baku masih menggunakan *microsoft excel*, Bagian gudang dalam mengecek ketersediaan stok bahan baku hanya melihat jumlah fisik yang ada digudang, sehingga dibutuhkan waktu yang lama dalam melakukan kegiatan tersebut. Bagian *Inventory Material* yang mempunyai wewenang dalam membuat laporan *purchase requisition* dan *purchase order* masih menemukan beberapa kekurangan dalam

menggunakan *microsoft excel* sehingga dalam membuat pengolahan data dan laporan kurang terintegrasi.

Agar efektifitas dan efisiensi PT Coca Cola Amatil Indonesia dapat tercapai, maka perlu menerapkan sistem secara terintegrasi. Informasi yang dihasilkan akan membantu bagian Produksi, *Inventory Material*, dan Gudang dalam membuat laporan menjadi lebih informatif dan mudah dipahami. Berdasarkan uraian-uraian diatas, maka penulisan tugas akhir ini diberi judul **“Rancang Bangun Sistem Informasi Pengendalian Persediaan Bahan Baku Dengan Menggunakan PHP 5.4.7 dan MySQL 5.5.27 Pada PT Coca Cola Amatil Indonesia “**.

1.2. Pokok Permasalahan

Berdasarkan latar belakang yang telah dipaparkan, maka dapat diidentifikasi permasalahan yang terjadi pada bagian *inventory material* line 13 PT Coca Cola Amatil Indonesia adalah sebagai berikut:

1. Proses pengecekan stok bahan baku masih melihat jumlah fisik yang ada digudang, sehingga membutuhkan waktu yang lama dalam pembuatan laporan stok persediaan bahan baku.
2. Proses pencatatan laporan *material list* bahan baku, *purchase requisition* dan *purchase order* masih menggunakan *microsoft excel*, sehingga dalam membuat pengolahan data dan laporan kurang terintegrasi.

1.3. Tujuan Penelitian

Adapun beberapa tujuan penelitian yang hendak dicapai dalam penelitian Tugas Akhir yang dilakukan di PT Coca Cola Amatil Indonesia adalah sebagai berikut:

1. Membangun sistem informasi pengendalian persediaan bahan baku dengan menggunakan bahasa pemrograman PHP 5.4.7 dan MySQL 5.5.27 agar laporan dapat dibuat dengan cepat.
2. Membuat laporan pengolahan data *material list*, *purchase requisition*, dan *purchase order* menjadi lebih terintegrasi dan mudah dipahami.

1.4. Batasan Masalah

Agar penulisan Tugas Akhir ini lebih terarah dan tidak meluas, maka perlu ditentukan batasan-batasan yang jelas sebagai berikut:

1. Penelitian dilakukan pada bagian *Inventory Material* di dalam PT.Coca Cola Amatil Cibitung Plant.
2. Hanya membahas pengendalian persediaan bahan baku pada bagian produksi line 13 Minuman Non Karbonasi *Minute Maid Pulpy Orange*.

1.5. Manfaat Tugas Akhir

Adapun manfaat yang dapat diperoleh dari penelitian Tugas Akhir ini adalah sebagai berikut:

1. Bagi penulis
Mengimplementasikan teori-teori yang telah diperoleh selama perkuliahan, dengan kenyataan yang ada dalam dunia usaha sehingga penulis dapat menambah pengetahuan tentang masalah-masalah yang dihadapi di dunia usaha yang terjadi pada perusahaan manufaktur.
2. Bagi perusahaan
Hasil penelitian ini dapat diimplementasikan di perusahaan untuk membantu kinerja perusahaan. Dengan adanya Sistem Informasi Pengendalian Persediaan ini dapat meningkatkan efisiensi serta efektivitas kerja dalam hal membuat laporan.
3. Bagi pihak lain
Hasil penelitian ini diharapkan dapat memberikan sumbangan pemikiran dan sebagai referensi bagi peneliti lain yang melakukan penelitian serupa.

1.6. Sistematika Penulisan

Agar lebih mempermudah perumusan dan penyelesaian masalah yang akan dibahas pada tugas akhir ini, maka diuraikan tahapan-tahapan dalam penyusunan laporan ini sebagai berikut:

BAB I PENDAHULUAN

Bab ini membahas tentang hal-hal yang bersifat umum dalam latar belakang, pokok permasalahan, batasan masalah, tujuan penelitian, manfaat tugas akhir, serta sistematika penulisan.

BAB II LANDASAN TEORI

Bab ini berisi tentang teori dari konsep dasar sistem, jenis sistem, konsep dasar informasi, kualitas informasi, siklus informasi, konsep dasar persediaan, pengendalian persediaan, pengembangan sistem, model prototype secara umum, evolutionary prototype, *flowchart*, *Unified Modelling Language* (UML), definisi HIPO, PHP 5.4.7 dan MySQL versi 5.5.27, serta teori-teori lain yang berhubungan dengan perancangan sistem yang digunakan selama proses penulisan tugas akhir ini.

BAB III METODOLOGI PENELITIAN

Bab ini membahas tentang metode pengumpulan data, serta langkah-langkah yang akan dilakukan dalam perumusan dan pemecahan masalah termasuk metodologi pengembangan sistem yang digunakan.

BAB IV PENGUMPULAN DAN PENGOLAHAN DATA

Bab ini berisi tentang hasil pengamatan yang telah dilakukan, seperti prosedur sistem pengendalian persediaan pengadaan bahan baku, menggambarkan sistem yang sedang berjalan menggunakan *flowchart* dan UML

BAB V ANALISIS DAN PEMBAHASAN

Bab ini berisi analisis rinci dari pengolahan data yang dilakukan, mulai dari analisis dengan *flowchart*, pemodelan *unified modeling language*, sampai dengan rancangan *interface* program.

BAB VI KESIMPULAN DAN SARAN

Bab ini berisikan kesimpulan dari penelitian yang dilakukan, serta mengemukakan saran-saran untuk perusahaan.

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Sistem

Secara umum, sistem dapat didefinisikan sebagai sekumpulan hal atau kegiatan atau elemen atau subsistem yang saling bekerja sama atau yang dihubungkan dengan cara-cara tertentu sehingga membentuk satu kesatuan untuk melaksanakan suatu fungsi guna mencapai suatu tujuan. Suatu sistem mempunyai karakteristik sebagai berikut (Sutanta, 2003):

1. Mempunyai komponen (*components*)

Komponen sistem adalah segala sesuatu yang menjadi bagian penyusun sistem. Komponen sistem dapat berupa benda nyata ataupun abstrak. Komponen sistem disebut sebagai subsistem, dapat berupa orang, benda, hal atau kejadian yang terlibat di dalam sistem.

2. Mempunyai batas (*boundary*)

Batas sistem diperlukan untuk membedakan satu sistem dengan sistem yang lain. Tanpa adanya batas sistem, maka sangat sulit untuk menjelaskan suatu sistem. Batas sistem akan memberikan batasan (*scope*) tinjauan terhadap sistem.

3. Mempunyai lingkungan (*environments*)

Lingkungan sistem adalah segala sesuatu yang berada di luar sistem. Lingkungan sistem dapat menguntungkan ataupun merugikan. Umumnya, lingkungan yang menguntungkan akan selalu dipertahankan untuk menjaga keberlangsungan sistem. Sedangkan lingkungan sistem yang merugikan akan diupayakan agar mempunyai pengaruh seminimal mungkin, bahkan jika mungkin ditiadakan.

4. Mempunyai penghubung atau antar muka (*interface*) antar komponen

Penghubung atau antar muka merupakan komponen sistem, yaitu segala sesuatu yang bertugas menjembatani hubungan antar komponen dalam

sistem. Penghubung atau antar muka merupakan sarana yang memungkinkan setiap komponen saling berinteraksi dan berkomunikasi dalam rangka menjalankan fungsi masing-masing komponen. Dalam dunia komputer, penghubung atau antar muka dapat berupa berbagai macam tampilan dialog layar monitor yang memungkinkan seseorang dapat dengan mudah mengoperasikan sistem aplikasi komputer yang digunakan.

5. Mempunyai masukan (*input*)

Masukan merupakan komponen sistem, yaitu segala sesuatu yang perlu dimasukan ke dalam sistem sebagai bahan yang akan diolah lebih lanjut untuk menghasilkan keluaran yang berguna. Dalam Sistem Informasi Manajemen, masukan disebut sebagai data.

6. Mempunyai pengolahan (*processing*)

Pengolah merupakan komponen sistem yang mempunyai peran utama mengolah masukan agar menghasilkan keluaran yang berguna bagi para pemakainya. Dalam Sistem Informasi Manajemen, pengolahan adalah berupa program aplikasi komputer yang dikembangkan untuk keperluan khusus. Program aplikasi tersebut mampu menerima masukan, mengolah masukan, dan menampilkan hasil olahan sesuai dengan kebutuhan para pemakai.

7. Mempunyai keluaran (*output*)

Keluaran merupakan komponen sistem yang berupa berbagai macam bentuk keluaran yang dihasilkan oleh komponen pengolah. Dalam Sistem Informasi Manajemen, keluaran adalah informasi yang dihasilkan oleh program aplikasi yang akan digunakan oleh para pemakai sebagai bahan pengambilan keputusan.

8. Mempunyai sasaran (*objectives*) dan tujuan (*goal*)

Setiap komponen dalam sistem perlu dijaga agar mampu mencapai sasaran dan tujuan sistem, sasaran berbeda dengan tujuan. Sasaran sistem adalah apa yang ingin dicapai oleh sistem untuk jangka waktu yang relatif pendek. Sedangkan tujuan merupakan kondisi atau hasil akhir yang ingin dicapai oleh sistem untuk jangka waktu yang panjang. Dalam hal ini,

sasaran merupakan hasil pada setiap tahapan tertentu yang mendukung upaya pencapaian tujuan.

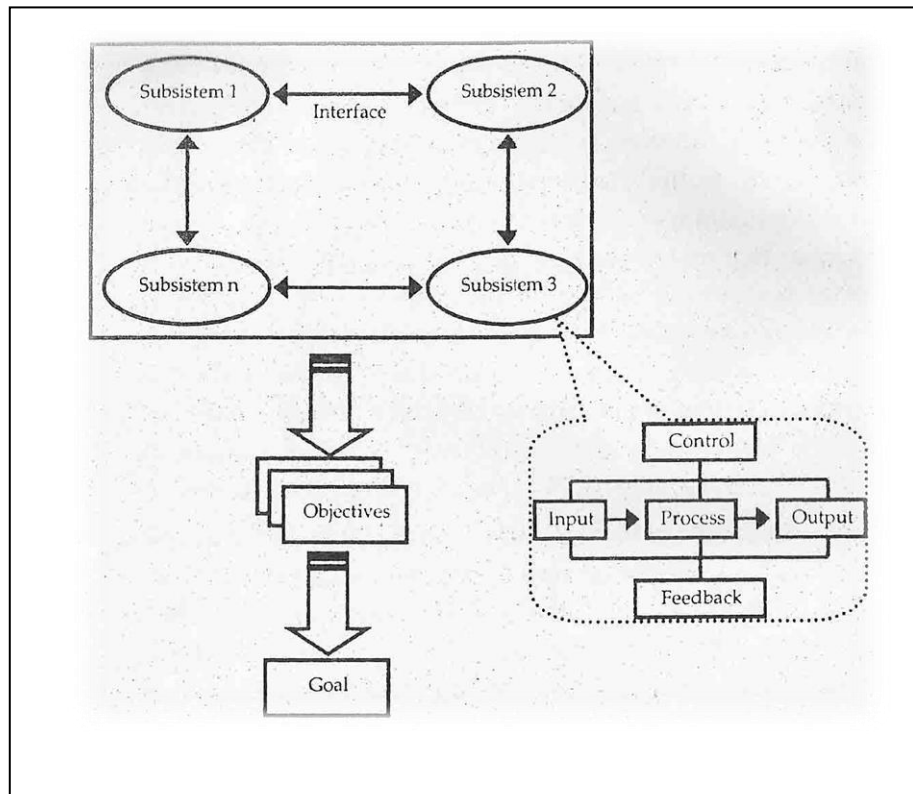
9. Mempunyai kendali (*control*)

Setiap komponen dalam sistem perlu selalu dijaga agar tetap bekerja sesuai dengan peran dan fungsinya masing-masing. Hal ini bisa dilakukan jika ada bagian yang berperan menjaganya, yaitu bagian kendali. Bagian kendali mempunyai peranan utama menjaga agar proses dalam sistem dapat berlangsung secara normal sesuai batasan yang telah ditetapkan sebelumnya. Dalam Sistem Informasi Manajemen, kendali dapat berupa validasi masukan, validasi proses, maupun validasi keluaran yang dapat dirancang dan dikembangkan secara terprogram.

10. Mempunyai umpan balik (*feedback*)

Umpan balik diperlukan oleh bagian kendali (*control*) sistem untuk mengecek terjadinya penyimpangan proses dalam sistem dan mengembalikannya ke dalam kondisi normal.

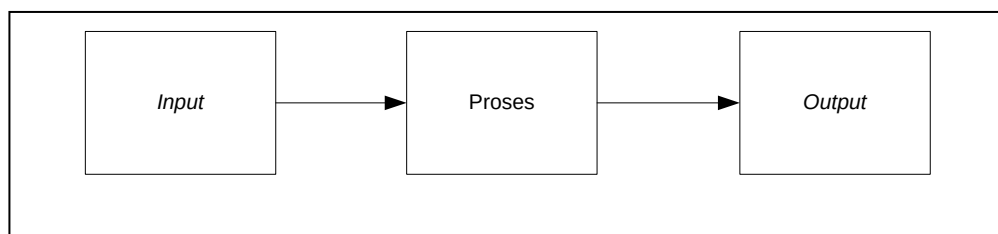
Keterkaitan antar komponen dan karakteristik suatu sistem di tunjukan oleh Gambar II.1. Dalam gambar tersebut, suatu subsistem berkaitan dengan subsistem lainnya dihubungkan oleh *interface*, membentuk satu kesatuan guna mencapai *objectives*, dan pada akhirnya diharapkan akan mencapai *goal*. Suatu subsistem bisa jadi memuat komponen *input*, *process*, dan *output* yang dikendalikan oleh bagian *control* yang melakukan kendali berdasarkan *feedback*. Dalam suatu sistem, subsistem 1 bisa juga berperan sebagai *input* bagi subsistem 2 yang berperan sebagai *process*.



Gambar II.1. Keterkaitan Antar Komponen dan Karakteristik Suatu Sistem

Sumber: Sutanta (2003)

Model umum suatu sistem adalah terdiri dari atas masukan (*input*), pengolah (*process*) dan keluaran (*output*), sebagaimana ditunjukkan oleh Gambar II.2 berikut:



Gambar II.2. Model Umum Suatu Sistem

Sumber: Sutanta (2003)

2.1.1. Jenis Sistem

Tinjauan tentang suatu sistem dapat diklasifikasikan dalam beberapa cara, yaitu (Sutanta, 2003):

1. Sistem fisis (*physical systems*) dan sistem abstrak (*abstract systems*)

Sistem fisis adalah sistem yang komponennya berupa benda nyata yang dapat dilihat atau dijamah oleh tangan manusia. Contoh sistem fisis adalah sistem perangkat keras (*hardware*) komputer yang antara lain terdiri atas, unit pusat pengolah (*Central Processing Unit* atau CPU), *memory*, *monitor*, *keyboard*, dan lainnya. Sedangkan sistem abstrak adalah sistem yang komponennya tidak dapat dilihat atau dijamah oleh tangan manusia. Contoh sistem abstrak adalah sistem operasi (*Operating Systems* atau OS) komputer yang terdiri atas sekumpulan instruksi dalam bahasa yang dipahami oleh mesin komputer. Umumnya suatu sistem terdiri atas gabungan komponen fisis dan abstrak yang saling bekerja sama.

2. Sistem alamiah (*natural systems*) dan sistem buatan manusia (*human made systems*)

Sistem alamiah adalah sistem yang keberadaannya terjadi secara alami atau natural tanpa campur tangan manusia. Sedangkan sistem buatan manusia ada sebagai hasil kerja manusia. Contoh sistem alamiah adalah sistem tata surya yang terdiri atas sekumpulan planet, gugusan bintang dan lainnya. Contoh sistem abstrak dapat berupa sistem komputer yang ada sebagai hasil karya teknologi yang dikembangkan oleh manusia.

3. Sistem tertentu (*deterministic systems*) dan sistem tidak tentu (*probabilistic systems*)

Sistem tertentu adalah sistem yang tingkah lakunya dapat ditentukan atau diprediksi sebelumnya. Sedangkan sistem tidak tertentu tingkah lakunya tidak dapat ditentukan atau diprediksi sebelumnya. Sistem aplikasi komputer merupakan contoh sistem yang tingkah lakunya dapat ditentukan sebelumnya. Program aplikasi komputer dirancang dan dikembangkan oleh manusia dengan menggunakan prosedur yang jelas, terstruktur, dan baik. Dengan demikian, untuk nilai-nilai masukan yang diberikan akan dapat diketahui nilai keluarannya secara pasti sebelumnya. Sedangkan sistem perekonomian dalam suatu negara termasuk klasifikasi sistem tidak tertentu, karena tidak diketahui dengan pasti apa yang akan

terjadi terhadap kondisi perekonomian tersebut apabila terjadi suatu kejadian tertentu. Hal ini bisa dipahami karena sistem perekonomian suatu bangsa dipengaruhi oleh banyak variabel atau hal, misal keadaan keamanan, politik, dan lainnya.

4. Sistem tertutup (*closed systems*) dan sistem terbuka (*open systems*)

Sistem tertutup merupakan sistem yang tingkah lakunya tidak dipengaruhi oleh lingkungan luarnya. Sebaliknya, sistem terbuka mempunyai perilaku yang dipengaruhi oleh lingkungannya. Dalam kenyataannya hampir tidak ada suatu sistem yang benar-benar tertutup. Yang ada adalah sistem yang relatif tertutup, yaitu sistem yang relatif tidak dipengaruhi oleh lingkungannya. Sistem aplikasi komputer merupakan contoh sistem relatif tertutup, karena tingkah laku sistem aplikasi komputer tidak dipengaruhi oleh kondisi yang terjadi di luar sistem. Sekalipun sistem aplikasi komputer akan berhenti apabila satu daya listrik ke komputer mengalami gangguan atau padam.

2.2. Konsep Dasar Informasi

Sumber dari informasi adalah data. Data merupakan bentuk jamak dari bentuk tunggal atau data-data. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Kejadian-kejadian adalah sesuatu yang terjadi pada saat tertentu (Jogiyanto, 2005). Fakta atau observasi mentah yang biasanya banyak data yang menjelaskan kegiatan tersebut. Informasi merupakan data yang telah diubah menjadi konteks yang berarti dan berguna bagi pemakai akhir tertentu (McLeod, 2004).

Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya (Jogiyanto, 2005).

Dari pengertian di atas informasi dapat disimpulkan sebagai kumpulan data yang telah diproses, diorganisasikan ke dalam bentuk yang sesuai dengan kebutuhan dan memiliki arti bagi penggunaannya.

2.2.1. Kualitas Informasi

Menurut Jogiyanto (2005) kualitas informasi yang diharapkan tergantung 4 (empat) hal pokok yaitu:

1. Akurat

Akurat mempunyai arti informasi yang dihasilkan harus bebas dari kesalahan-kesalahan yang tidak biasa, tidak menyesatkan dan menceminkan maksudnya.

2. Tepat waktu

Tepat waktu berarti informasi yang disampaikan ke penerima tidak terlambat, karena informasi adalah landasan untuk mengambil suatu keputusan. Untuk itu diperlukan suatu teknologi untuk dan mengirim dengan cepat dan tepat agar informasi tidak usang.

3. Relevan

Berarti informasi mempunyai manfaat dan berguna bagi pemakainya. Karena batas relevansi seseorang berbeda, maka informasi bisa dikatakan berguna jika benar-benar berguna dan dibutuhkan pemakainya.

4. Aman

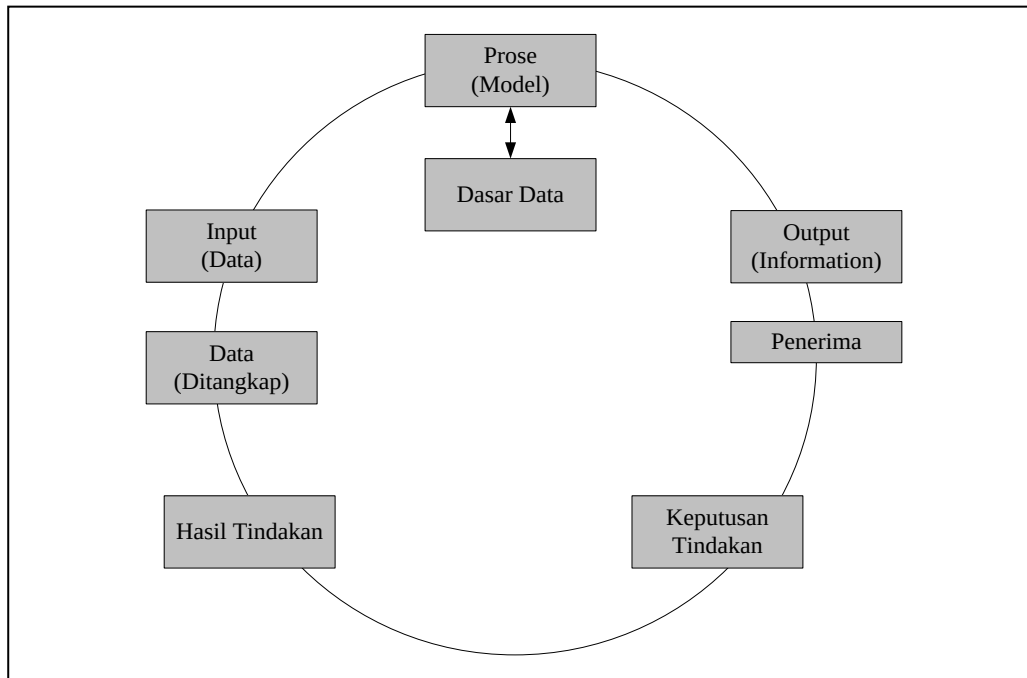
Aman berarti informasi harus terbebas dari penyadapan oleh pihak orang yang tidak berwenang dalam penggunaan informasi tersebut.

2.2.2. Siklus Informasi

Data merupakan bentuk yang masih mentah, belum dapat bercerita banyak sehingga perlu diolah lebih lanjut. Data diolah melalui suatu model untuk menghasilkan informasi. Data dapat berbentuk simbol-simbol semacam huruf-huruf atau alphabet, angka-angka, bentuk-bentuk suara, sinyal-sinyal, gambar-gambar dan sebagainya.

Data yang diolah melalui suatu model menjadi informasi, kemudian penerima menerima informasi tersebut, membuat suatu keputusan dan melakukan tindakan, yang berarti menghasilkan suatu tindakan yang lain yang akan membuat sejumlah data kembali. Data tersebut akan ditangkap sebagai *input*, diproses kembali lewat suatu model dan seterusnya membentuk suatu siklus. Siklus ini oleh John Burch disebut dengan siklus informasi (*information cycle*) atau ada

yang menyebutnya dengan istilah siklus pengolahan data (*data processing cycle*) yang dapat dilihat pada Gambar II.3 berikut.



Gambar II.3. Siklus Informasi

Sumber: Jogiyanto (2005)

2.3. Pengertian Persediaan

Setiap perusahaan apakah itu perusahaan perdagangan atau pabrik serta perusahaan jasa selalu mengadakan persediaan, karena itu persediaan sangat penting, tanpa adanya persediaan para pengusaha yang mempunyai perusahaan-perusahaan tersebut akan dihadapkan pada resiko-resiko yang dihadapi, misalnya; pada sewaktu-waktu perusahaan tidak dapat memenuhi keinginan pelanggan yang memerlukan atau meminta barang atau jasa yang dihasilkan. Hal tersebut dapat terjadi karena disetiap perusahaan tidak selamanya barang-barang atau jasa-jasa tersedia setiap saat, yang berarti pengusaha akan kehilangan kesempatan memperoleh keuntungan yang seharusnya didapatkan.

Begitu pentingnya persediaan sehingga merupakan elemen utama terbesar dari modal kerja yang merupakan aktiva yang selalu dalam keadaan berputar dimana secara terus-menerus mengalami perubahan.

Persediaan menurut Assauri (2010) adalah suatu aktiva yang meliputi barang-barang milik perusahaan yang dimaksud untuk dijual dalam satu periode usaha yang normal atau persediaan barang baku yang menunggu penggunaannya dalam suatu proses produksi.

Pada dasarnya persediaan mempermudah atau memperlancar jalannya operasi perusahaan pabrik yang harus dilakukan secara berturut-turut untuk memproduksi barang-barang, serta selanjutnya menyampaikan pada pelanggan atau konsumen. Persediaan memungkinkan produk-produk yang dihasilkan pada tempat yang jauh dari pelanggan atau sumber bahan mentah. Dengan adanya persediaan produksi tidak perlu dilakukan khusus buat konsumsi atau sebaliknya tidak perlu dikonsumsi didesak supaya sesuai dengan kepentingan produksi. Adapun alasan diperlukannya persediaan oleh suatu perusahaan menurut Assauri (2010) adalah sebagai berikut:

1. Dibutuhkannya waktu untuk menyelesaikan operasi produksi untuk memindahkan produk dari satu tingkat proses yang lain yang disebut persediaan dalam proses dan pemindahan.
2. Alasan organisasi untuk memungkinkan suatu unit atau bagian membuat skedul operasinya secara bebas tidak tergantung dari yang lainnya.

Sedangkan persediaan yang diadakan mulai dari yang bentuk bahan mentah sampai dengan barang jadi antara lain berguna untuk dapat: Menurut Assauri (2010):

1. Menghilangkan resiko keterlambatan datangnya barang atau bahan-bahan yang dibutuhkan perusahaan.
2. Menghilangkan resiko dari material yang dipesan tidak baik sehingga harus dikembalikan.
3. Untuk menumpuk bahan-bahan yang dihasilkan secara musiman sehingga dapat digunakan bila bahan itu tidak ada dalam pasaran.

4. Mempertahankan stabilitas operasi perusahaan atau menjamin kelancaran arus produksi
5. Mencapai penggunaan mesin yang optimal.
6. Memberikan pelayanan (*service*) kepada pelanggan dengan sebaik-baiknya dimana keinginan pelanggan pada suatu waktu dapat dipenuhi adalah memberikan jaminan tetap tersedianya barang jadi tersebut
7. Membuat pengadaan atau produksi tidak perlu sesuai dengan penggunaan atau penjualannya.

2.3.1. Jenis-Jenis Persediaan

Dalam merencanakan dan mengendalikan persediaan terdapat jenis-jenis persediaan yang digunakan. Dilihat dari fungsinya persediaan menurut Assauri (2010) adalah sebagai berikut:

1. *Batch Stock* atau *Lot size Inventory* yaitu persediaan yang diadakan karena kita membeli atau membuat bahan-bahan atau barang-barang dalam jumlah yang lebih besar daripada jumlah yang dibutuhkan pada saat itu. Adapun keuntungan yang diperoleh dari adanya *Lot Size Inventory* adalah sebagai berikut:
 - a. Memperoleh potongan harga pada harga pembelian
 - b. Memperoleh efisiensi produksi (*manufacturing economis*) karena adanya operasi atau “production run” yang lebih lama.
 - c. Adanya penghematan didalam biaya angkutan.
2. *Fluctuation Stock* adalah persediaan yang diadakan untuk menghadapi fluktuasi permintaan konsumen yang tidak dapat diramalkan.
3. *Anticipation stock* adalah persediaan yang diadakan untuk menghadapi fluktuasi permintaan yang dapat diramalkan, berdasarkan pola musiman yang terdapat dalam satu tahun dan untuk menghadapi penggunaan atau penjualan permintaan yang meningkat.

Sedangkan persediaan dilihat dari jenis atau posisi menurut Assauri (2010) dapat dibedakan sebagai berikut:

1. Persediaan bahan baku (*Raw Material stock*) yaitu persediaan dari barang-barang berwujud yang digunakan dalam proses produksi, barang mana dapat diperoleh dari sumber-sumber alam ataupun dibeli dari supplier atau perusahaan yang menghasilkan bahan baku bagi perusahaan pabrik yang menggunakan nya.
2. Persediaan bagian produk (*Purchased part*) yaitu persediaan barang-barang yang terdiri dari part atau bagian yang diterima dari perusahaan lain, yang dapat secara langsung diassembling dengan part lain, tanpa melalui proses produksi sebelumnya.
3. Persediaan bahan-bahan pembantu atau barang-barang perlengkapan (*Supplies stock*) yaitu persediaan barang-barang atau bahan-bahan yang diperlukan dalam proses produksi untuk membantu berhasilnya produksi atau yang dipergunakan dalam bekerjanya suatu perusahaan, tetapi tidak merupakan bagian atau komponen dari barang jadi.
4. Persediaan barang setengah jadi atau barang dalam proses (*work in process/progress stock*) yaitu persediaan barang-barang yang keluar dari tiap-tiap bagian dalam satu pabrik atau bahan-bahan yang telah diolah menjadi suatu bentuk, tetapi lebih perlu diproses kembali untuk kemudian menjadi barang jadi.
5. Persediaan barang jadi (*Finished goods stock*) yaitu barang-barang yang telah selesai diproses atau diolah dalam pabrik dan siap untuk dijual kepada pelanggan atau perusahaan lain.

2.3.2. Faktor-Faktor Yang Mempengaruhi Persediaan

Meskipun persediaan akan memberikan banyak manfaat bagi perusahaan, namun perusahaan tetap hati-hati dalam menentukan kebijakan persediaan. Persediaan membutuhkan biaya investasi dan dalam hal ini menjadi tugas bagi manajemen untuk menentukan investasi yang optimal dalam persediaan. Masalah persediaan merupakan masalah pembelanjaan aktif, dimana perusahaan

menemukan dana yang dimiliki dalam persediaan dengan cara yang seefektif mungkin.

Untuk melangsungkan usahanya dengan lancar maka kebanyakan perusahaan merasakan perlunya persediaan. Menurut Riyanto (2009) Besar kecilnya persediaan yang dimiliki oleh perusahaan ditentukan oleh beberapa faktor antara lain:

1. Volume yang dibutuhkan untuk melindungi jalannya perusahaan terhadap gangguan kehabisan persediaan yang akan menghambat atau mengganggu jalannya produksi.
2. Volume produksi yang direncanakan, dimana volume produksi yang direncanakan itu sendiri sangat tergantung kepada volume sales yang direncanakan.
3. Besar pembelian bahan mentah setiap kali pembelian untuk mendapatkan biaya pembelian yang minimal.
4. Estimasi tentang fluktuasi harga bahan mentah yang bersangkutan diwaktu-waktu yang akan datang.
5. Peraturan-peraturan pemerintah yang menyangkut persediaan material.
6. Harga pembelian bahan mentah.
7. Biaya penyimpanan dan resiko penyimpanan di gudang.
8. Tingkat kecepatan material menjadi rusak atau turun kualitasnya.

2.3.3. Peranan Perencanaan dan Pengendalian Persediaan

Perencanaan dan pengendalian merupakan bagian dari manajemen persediaan. Pengendalian adalah suatu tindakan agar aktifitas dilakukan dengan sebaik-baiknya sesuai dengan rencana yang telah ditetapkan. Pengendalian tanpa perencanaan adalah sia-sia dan perencanaan tanpa pengendalian merupakan tindakan yang tidak efektif.

Secara umum dapat diformulasikan disini bahwa arti dari perencanaan dan pengendalian bahan baku adalah suatu kegiatan memperkirakan kebutuhan persediaan bahan baku, baik secara kualitatif maupun kuantitatif. Agar perusahaan

dapat beroperasi seperti yang direncanakan, jadi singkatnya bahwa arti dari perencanaan dan pengendalian persediaan bahan baku, persediaan bahan setengah jadi dan persediaan barang jadi. Secara keseluruhan diartikan sebagai upaya menentukan besarnya tingkat persediaan dan mengendalikannya dengan efisien dan efektif.

Untuk menentukan pengendalian persediaan bahan baku yang efektif maka diperlukan tujuan perencanaan yang efektif pula dan merupakan kegiatan pengendalian (*Controlling*). Adapun tujuan perencanaan bahan baku adalah:

- a. Agar jumlah persediaan bahan yang disediakan tidak terlalu sedikit juga terlalu banyak, artinya dalam jumlah yang cukup efisien dan efektif.
- b. Operasi perusahaan khususnya proses produksi dapat berjalan secara efisien dan efektif.
- c. Implikasi penyediaan bahan yang efisien demi untuk kelancaran proses produksi, berarti harus disediakan investasi sejumlah modal dalam jumlah yang memadai.

Untuk mengatur tingkat persediaan dalam jumlah, mutu, dan waktu yang tepat. Maka diperlukan pengendalian persediaan bahan yang efektif dan efisien, untuk itu penulis menyajikan pengertian pengendalian persediaan bahan baku.

Oleh karena itu perusahaan harus mengadakan suatu tingkat persediaan yang tepat karena bila persediaan terlalu berlebihan berarti lebih banyak uang atau modal yang tertanam dan biaya-biaya yang ditimbulkan dari persediaan tersebut besar jumlah dan bila persediaan terlalu kecil akan mengganggu kelancaran dari kegiatan produksi perusahaan. Untuk menentukan pengendalian persediaan maka harus memenuhi persyaratan-persyaratan menurut Assauri (2010) adalah sebagai berikut:

1. Terdapatnya gudang yang cukup luas dan teratur dengan pengaturan tempat bahan atau barang yang tetap dan identifikasi bahan atau barang tertentu.
2. Sentralisasi kekuasaan dan tanggung jawab pada satu orang dapat dipercaya terutama penjaga gudang.

3. Suatu sistem pencatatan dan pemeriksaan atas penerimaan bahan atau barang.
4. Pengawasan mutlak atas pengeluaran bahan atau barang.

2.4. Pengembangan Sistem

Pengembangan sistem dapat berarti menyusun suatu sistem yang baru untuk menggantikan sistem yang lama secara keseluruhan atau memperbaiki sistem yang telah ada. Sistem yang lama perlu diperbaiki oleh karena beberapa hal, yaitu (Sutabri, 2003):

1. Munculnya masalah pada sistem yang lama.
2. Untuk meraih kesempatan.
3. Adanya instruksi.

Proses perancangan atau pengembangan sistem informasi, mulai dari konsep sampai dengan implementasinya disebut dengan istilah *Software Development Life Cycle* atau sering disebut juga *System Development Life Cycle* (SDLC) adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya berdasarkan *best practice* atau cara-cara yang sudah teruji baik (Rosa dan Shalahuddin, 2011).

Menurut Rosa dan Shalahuddin (2011), tahapan-tahapan yang ada pada SDLC secara global adalah sebagai berikut:

1. Inisiasi (*initiation*)
Tahap ini biasanya ditandai dengan pembuatan proposal proyek perangkat lunak.
2. Pengembangan konsep sistem (*system concept development*)
Mendefinisikan lingkup konsep termasuk dokumen lingkup sistem analisis manfaat biaya, manajemen rencana dan pembelajaran kemudahan sistem.
3. Perencanaan (*planning*)

Mengembangkan rencana manajemen proyek dan dokumen perencanaan lainnya. Menyediakan dasar untuk mendapatkan sumber daya (*resources*) yang dibutuhkan untuk memperoleh solusi.

4. Analisis kebutuhan (*requirements analysis*)

Menganalisis kebutuhan pemakai sistem perangkat lunak (*user*) dan mengembangkan kebutuhan *user*. Membuat dokumen kebutuhan fungsional.

5. Desain (*design*)

Mentransformasikan kebutuhan detail menjadi kebutuhan yang sudah lengkap, dokumen desain sistem fokus pada bagaimana dapat memenuhi fungsi-fungsi yang dibutuhkan.

6. Pengembangan (*development*)

Mengkonversi desain ke sistem informasi yang lengkap termasuk bagaimana memperoleh dan melakukan instalasi lingkungan sistem yang dibutuhkan, membuat basis data dan mempersiapkan prosedur kasus pengujian, mempersiapkan berkas atau *file* pengujian, pengkodean, pengkompilasian, memperbaiki dan membersihkan program serta peninjauan pengujian.

7. Integrasi dan pengujian (*integration and test*)

Mendemonstrasikan sistem perangkat lunak bahwa telah memenuhi kebutuhan yang dispesifikasikan pada dokumen kebutuhan fungsional. Dengan diarahkan oleh staf penjamin kualitas (*quality assurance*) dan user sehingga menghasilkan laporan analisis pengujian.

8. Implementasi (*implementation*)

Termasuk pada persiapan implementasi, implementasi perangkat lunak pada lingkungan produksi (lingkungan luar *user*) dan menjalankan resolusi dari permasalahan yang teridentifikasi dari fase integrasi dan pengujian.

9. Operasi dan pemeliharaan (*operations and maintenance*)

Mendesripsikan pekerjaan untuk mengoperasikan dan memelihara sistem informasi pada lingkungan produksi (lingkungan pada *user*), termasuk implementasi akhir dan masuk pada proses peninjauan.

10. Disposisi (*disposition*)

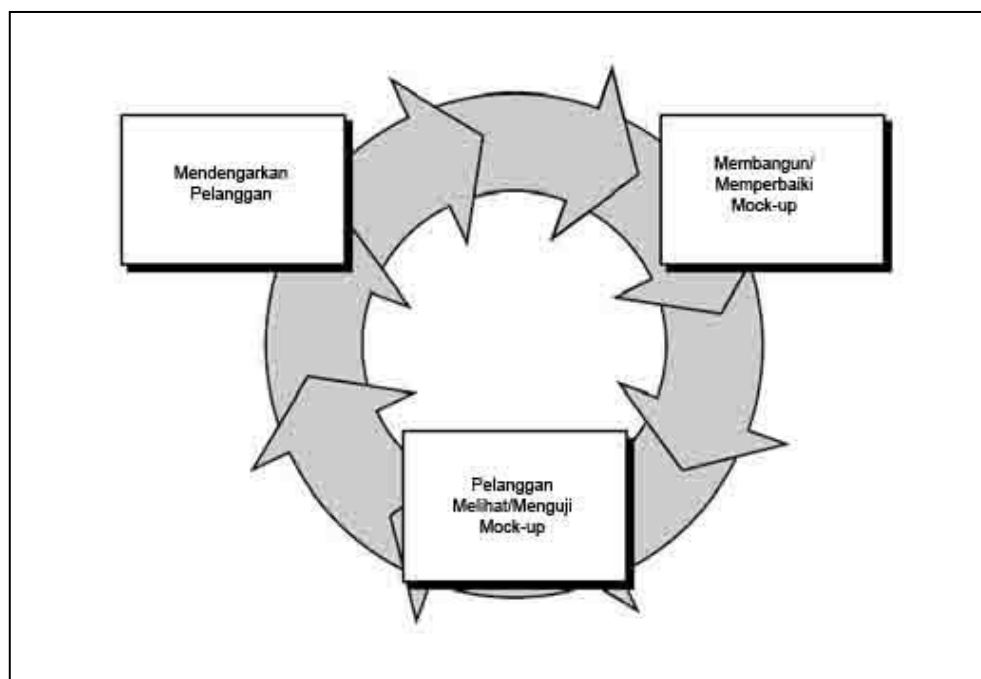
Mendeskripsikan aktifitas akhir dari pengembangan sistem dan membangun data yang sebenarnya sesuai dengan aktifitas user

2.4.1. Model Prototype Secara Umum

Model prototype dapat digunakan untuk menyambungkan ketidakpahaman pelanggan mengenai hal teknis yang memperjelas spesifikasi kebutuhan yang diinginkan pelanggan kepada pengembang perangkat lunak (Rosa dan Shalahuddin, 2011).

Tahapan-tahapan pada model prototype (*prototype model*) adalah sebagai berikut (Rosa dan Shalahuddin, 2011):

1. Mengumpulkan kebutuhan pelanggan terhadap perangkat lunak yang akan dibuat.
2. Membuat prototype agar pelanggan lebih terbayang dengan apa yang sebenarnya diinginkan. Program prototype biasanya merupakan program yang belum jadi.
3. Program prototype selanjutnya dievaluasi oleh pelanggan atau user sampai ditemukan spesifikasi yang sesuai dengan keinginan pelanggan atau user.



Gambar II.4. Ilustrasi Model Prototipe

Sumber: Rosa dan Shalahuddin (2011)

Mock-up adalah sesuatu yang digunakan sebagai model desain yang digunakan untuk mengajar, demonstrasi, evaluasi desain, promosi atau keperluan lain. Sebuah *mock-up* disebut sebagai prototipe perangkat lunak jika menyediakan atau mampu mendemonstrasikan sebagian besar fungsi sistem perangkat lunak dan memungkinkan pengujian desain sistem perangkat lunak. Iterasi terjadi pada pembuatan prototipe sampai sesuai dengan keinginan pelanggan atau *user* (Rosa dan Shalahuddin, 2011).

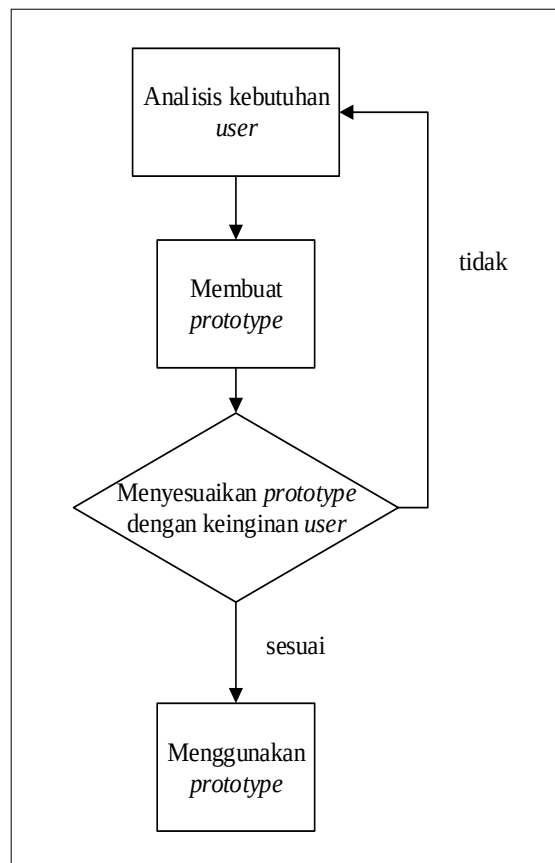
Kelemahan model prototipe adalah sebagai berikut (Rosa dan Shalahuddin, 2011):

1. *User* dapat sering mengubah-ubah atau menambah spesifikasi kebutuhan karena menganggap aplikasi sudah dengan cepat dikembangkan, karena adanya iterasi ini dapat menyebabkan pengembang banyak mengalah dengan *user* karena perubahan atau penambahan spesifikasi kebutuhan perangkat lunak.
2. Pengembang lebih sering mengambil kompromi dengan pelanggan untuk mendapatkan prototipe dengan waktu yang cepat sehingga pengembang lebih sering melakukan segala cara (tanpa idealis) guna menghasilkan prototipe untuk didemonstrasikan. Hal ini dapat menyebabkan kualitas perangkat lunak yang kurang baik atau bahkan menyebabkan iteratif tanpa akhir.

2.4.2. *Evolutionary Prototype*

Evolutionary prototype yaitu, *prototype* yang secara terus menerus dikembangkan hingga *prototype* tersebut memenuhi fungsi dan prosedur yang dibutuhkan oleh sistem. Pada pendekatan evolusioner, suatu *prototype* dibangun berdasarkan pada kebutuhan dan pemahaman secara umum. *Prototype* kemudian diubah dan dievolusikan dari pada dibuang. *Prototype* yang dibuang biasanya digunakan dengan aspek sistem yang dimengerti secara luas dan dibangun atas

kekuatan tahapan *evolutionary prototype* (McLeod, 2008). *Evolutionary Prototype Model* dapat dilihat pada Gambar II.5 berikut.



Gambar II.5. *Evolutionary Prototype Model*

Sumber: McLeod (2008)

1. Analisis kebutuhan user, pengembang dan user atau pemilik sistem melakukan diskusi dimana user atau pemilik sistem menjelaskan kepada pengembang tentang kebutuhan sistem yang mereka inginkan.
2. Membuat prototype, pengembang membuat prototype dari sistem yang telah dijelaskan oleh user atau pemilik sistem.
3. Menyesuaikan prototype dengan keinginan user atau pemilik sistem, pengembang menanyakan kepada user atau pemilik sistem tentang prototype yang sudah dibuat, apakah sesuai atau tidak dengan kebutuhan sistem.

4. Menggunakan prototype, sistem mulai dikembangkan dengan prototype yang sudah dibuat.

2.5. **Flowchart**

Bagan alir (*flowchart*) adalah bagan yang menggambarkan urutan instruksi proses dan hubungan satu proses dengan proses lainnya menggunakan simbol-simbol tertentu (Yakub, 2012).

Ada beberapa bagan alir (*flowchart*) yang biasa digunakan, antara lain (Jogiyanto, 2005):

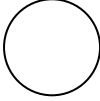
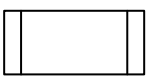
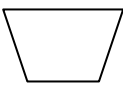
1. *Flowchart* sistem, yaitu bagan yang menunjukkan arus pekerjaan secara keseluruhan dari sistem yang menjelaskan urutan dari prosedur-prosedur yang ada di dalam sistem.
2. *Flowchart* dokumen, yaitu *flowchart* yang menunjukkan arus dari laporan dan formulir.
3. *Flowchart* skematik, yaitu bagan yang menggambarkan prosedur di dalam sistem. *Flowchart* ini menggunakan gambar-gambar komputer dan peralatan lainnya dalam pembuatan *flowchart* dengan tujuan untuk memudahkan komunikasi kepada orang yang kurang paham dengan simbol-simbol *flowchart*.
4. *Flowchart* program, yaitu bagan yang menjelaskan secara rinci langkah-langkah dari proses program dan dibuat atas dasar *flowchart* sistem.
5. *Flowchart* proses merupakan bagan alir yang banyak digunakan di teknik industri dan berguna bagi analisis sistem untuk menggambarkan proses dalam suatu prosedur.

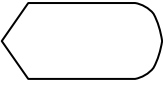
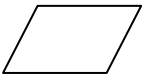
Simbol-simbol yang terdapat pada *Flowchart* dapat dilihat pada Tabel II.1 berikut.

Tabel II.1 Simbol-simbol *Flowchart*

Simbol	Nama	Keterangan
	Dokumen	Digunakan untuk semua jenis dokumen yang merupakan formulir untuk merekam transaksi.

Tabel II.1 Simbol-simbol *Flowchart* (Lanjutan)

Simbol	Nama	Keterangan
	Dokumen rangkap	Menggambarkan dokumen asli dan tembusannya.
	Penghubung pada halaman yang sama	Simbol penghubung yang memungkinkan aliran dokumen berhenti di suatu lokasi pada halaman tertentu dan kembali berjalan pada halaman yang sama.
	Proses terdefinisi	Menunjukkan proses rincian
	Penghubung pada halaman yang berbeda	Untuk menggambarkan bagan alir dokumen suatu sistem diperlukan lebih dari satu halaman.
	Kegiatan manual	Untuk menggambarkan kegiatan manual seperti menerima <i>order</i> , mengisi formulir, membandingkan dan lain-lain.
	Arsip sementara	Menunjukkan tempat penyimpanan dokumen.
	Arsip permanen	Menunjukkan tempat penyimpanan dokumen secara permanen yang tidak akan diproses lagi.
	Proses	Simbol proses yang digunakan untuk mewakili suatu proses dengan komputer.

	Keputusan	Menggambarkan keputusan yang harus dibuat dalam proses pengolahan data.
	Tampilan <i>Output</i>	Menunjukkan <i>output</i> yang ditampilkan pada monitor.
	<i>Input/Output</i>	Menggambarkan data <i>input/output</i> dari proses

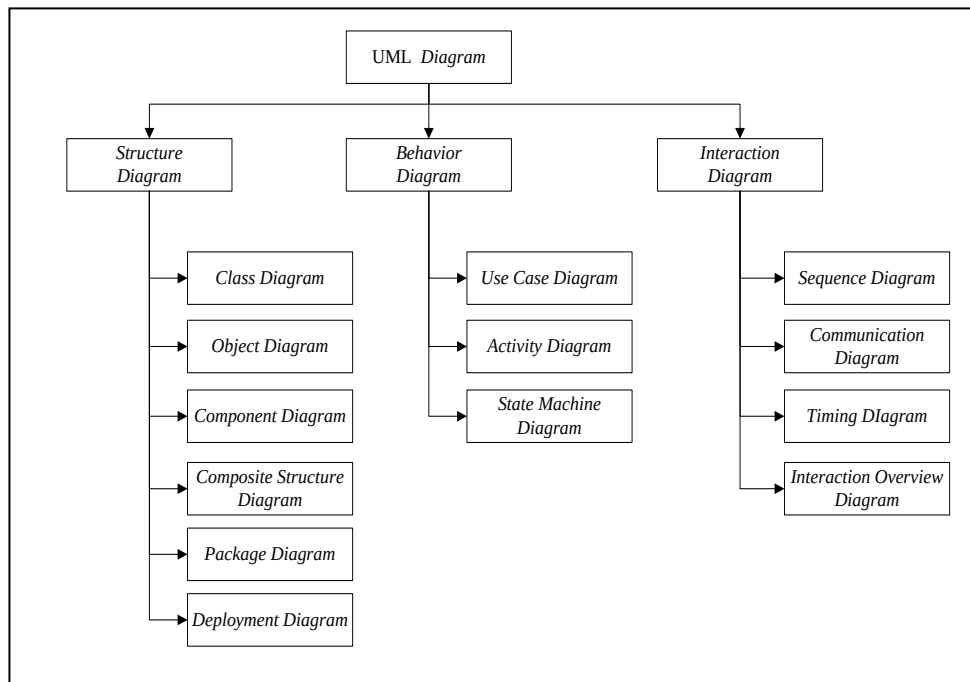
Sumber: Jogiyanto (2005)

2.6. *Unified Modelling Language (UML)*

Menurut Munawar (2005), UML (*Unified Modelling Language*) merupakan salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi objek. Hal ini dikarenakan UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi dan mengkomunikasikan rancangan mereka dengan yang lain. Adapun tujuan dari pemodelan UML adalah sebagai berikut:

1. Memberikan model yang siap pakai, bahasa pemodelan visual yang ekspresif untuk mengembangkan dan saling menukar model dengan mudah dan dimengerti secara umum.
2. Memberikan bahasa pemodelan yang bebas dari berbagai bahasa pemrograman dan proses rekayasa.
3. Menyatukan praktik-praktik terbaik yang terdapat dalam pemodelan.

UML terdiri dari 13 macam diagram yang dikelompokkan dalam tiga kategori, pembagian kategori tersebut dapat dilihat pada Gambar II.6.



Gambar II.6 Klasifikasi Diagram UML

(Sumber: Rosa dan Shalahudin, 2011)

Berikut ini penjelasan dari pembagian kategori tersebut, antara lain:

1. *Structure diagram*, yaitu kumpulan diagram yang menggambarkan suatu struktur statis dari sistem yang dimodelkan.
2. *Behavior diagram*, yaitu kumpulan diagram yang menggambarkan proses sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
3. *Interaction diagram*, yaitu kumpulan diagram yang menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar sub sistem pada suatu sistem.

2.6.1. *Use Case Diagram*

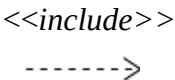
Menurut Rosa dan Shalahuddin (2011), *use case diagram* merupakan pemodelan untuk perilaku (*behavior*) sistem informasi yang akan dibuat. *Use case* menggambarkan interaksi antara sistem dengan eksternal sistem dan pengguna. Dengan kata lain, secara grafis menggambarkan siapa yang akan menggunakan

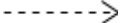
sistem dan dengan cara apa pengguna mengharapkan untuk berinteraksi dengan sistem. Simbol-simbol pada *use case diagram* dapat dilihat pada Tabel II.2.

Tabel II.2 Simbol-simbol *Use Case Diagram*

Simbol	Nama	Keterangan
	Aktor/ <i>Actor</i>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambaran orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
	<i>Use Case</i>	Fungsionalitas yang disediakan oleh sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i> .

Tabel II.2 Simbol-simbol *Use Case Diagram* (Lanjutan)

Simbol	Nama	Keterangan
	Asosiasi/ <i>Association</i>	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
	<i>Include</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.

	Generalisasi/ <i>Generalization</i>	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya. Arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum).
	<<extend>> Ekstensi/ <i>Extend</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri tanpa <i>use case</i> tambahan itu, mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek.

(Sumber: Rosa dan Shalahudin, 2011)

2.6.2. Activity Diagram

Menurut Rosa dan Shalahuddin (2011), *activity diagram* menggambarkan aliran kerja atau aktivitas dari sebuah sistem yang ada pada perangkat lunak. Adapun yang perlu diperhatikan disini adalah bahwa *activity diagram* menggambarkan aktivitas sistem bukan apa yang dilakukan aktor. Adapun simbol-simbol pada *activity diagram* dapat dilihat pada Tabel II.3.

Tabel II.3 Simbol-simbol Activity Diagram

Simbol	Nama	Keterangan
	Status Awal	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
	Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
	Percabangan/ <i>decision</i>	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.

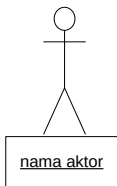
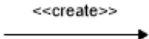
	Penggabungan/ <i>join</i>	Asosiasi penggabungan dimana lebih dari satu aktivitas yang digabungkan menjadi satu.
	<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

(Sumber: Rosa dan Shalahudin, 2011)

2.6.3. *Sequence Diagram*

Menurut Rosa dan Shalahuddin (2011), *sequence diagram* yaitu diagram yang menggambarkan perilaku objek pada *use case* dengan mendeskripsikan waktu hidup objek (*life line*) dan *message* yang dikirimkan dan diterima antar objek. Untuk menggambar *sequence diagram* terlebih dahulu harus mengetahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Simbol-simbol pada *sequence diagram* dapat dilihat pada Tabel II.4.

Tabel II.4 Simbol-simbol *Sequence Diagram*

Simbol	Nama	Keterangan
	Aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi, jadi simbol dari aktor ini adalah gambaran orang.
	Garis Hidup (<i>lifeline</i>)	Menyatakan kehidupan suatu objek.
	Waktu Aktif	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya.
	Pesan tipe <i>Create</i>	Menyatakan suatu objek membuat objek lain, arah panah mengarah pada objek yang dibuat.

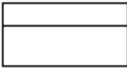
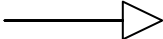
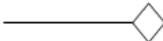
	Pesan tipe <i>Call</i>	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri. Arah panah mengarah pada objek yang memiliki operasi/metode.
	Pesan tipe <i>Send</i>	Menyatakan bahwa suatu objek mengirimkan data/masukan/ informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
	Pesan tipe <i>Return</i>	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
	Pesan tipe <i>Destroy</i>	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i> .

(Sumber: Rosa A. S. dan Shalahudin, 2011)

2.6.4. *Class Diagram*

Menurut Rosa dan Shalahuddin (2011), *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. *Class diagram* dibuat agar pembuat program atau *programmer* membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron. Simbol-simbol pada *class diagram* dapat dilihat pada Tabel II.5

Tabel II.5 Simbol-simbol *Class Diagram*

Simbol	Nama	Keterangan
	Kelas/ <i>Class</i>	Kelas pada struktur sistem.
	Asosiasi/ <i>Association</i>	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
	Antarmuka/ <i>Interface</i>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
	Asosiasi Berarah/ <i>Directed Association</i>	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
	Generalisasi/ <i>Generalization</i>	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum-khusus).
	Agregasi/ <i>Aggregation</i>	Relasi antarkelas dengan makna semua-bagian (<i>whole-part</i>).

(Sumber: Rosa dan Shalahudin, 2011)

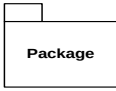
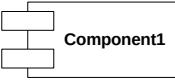
2.6.5. *Deployment Diagram*

Menurut Rosa dan Shalahuddin (2011), *deployment diagram* menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi dan dapat juga digunakan untuk memodelkan hal-hal antara lain:

1. Sistem tambahan (*embedded system*) yang menggambarkan rancangan *device*, *node*, dan *hardware*.
2. Sistem *client/server*
3. Sistem terdistribusi murni
4. Rekayasa ulang aplikasi

Simbol-simbol pada *deployment diagram* dapat dilihat pada Tabel II.6.

Tabel II.6 Simbol-simbol *Deployment Diagram*

Simbol	Nama	Keterangan
	<i>Package</i>	Merupakan sebuah bungkus dari satu atau lebih <i>node</i>
	<i>Node</i>	Biasanya mengacu pada perangkat keras dan perangkat lunak yang tidak dibuat sendiri, jika di dalam <i>node</i> disertakan komponen untuk konsistensi rancangan maka komponen yang diikutsertakan sesuai dengan komponen yang telah didefinisikan pada diagram komponen.
	Komponen	Komponen sistem
	Kebergantungan/ <i>n/ dependency</i>	Kebergantungan antar komponen, arah panah mengarah pada komponen yang dipakai.
	<i>Link</i>	Relasi antar komponen.

(Sumber: Rosa dan Shalahudin, 2011)

2.7. Kamus Data

Menurut Jogiyanto (2005), kamus data adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Dengan menggunakan kamus data diharapkan, analisis sistem dapat mendefinisikan data yang mengalir dalam sistem dengan lengkap. Kamus data dapat berfungsi membantu pelaku sistem untuk mengartikan aplikasi secara detail dan mengorganisasi semua elemen data yang digunakan di dalam sistem secara persis sehingga pemakai dan penganalisis sistem mempunyai dasar pengertian yang sama tentang masukan, keluaran, penyimpanan dan proses. Untuk maksud keperluan tersebut maka kamus data harus memuat hal-hal berikut:

1. Nama Arus Data

Nama arus data dibuat berdasarkan arus data yang mengalir maka nama dari arus data juga harus dicatat di kamus data, sehingga mereka yang

memerlukan penjelasan lebih lanjut tentang arus data dapat langsung mencarinya.

2. Alias

alias atau nama lain dari data dapat dituliskan bila nama lain ini ada. Alias perlu ditulis karena data yang sama mempunyai nama yang berbeda untuk orang atau departemen satu dengan yang lainnya.

3. Bentuk data

Bentuk dari data yang mengalir dapat berupa :

- a. Dokumen dasar atau formulir
- b. Dokumen hasil cetakan komputer
- c. Laporan tercetak
- d. Tampilan dilayar monitor
- e. Variabel
- f. Parameter
- g. *Field*

4. Arus data

Arus data menunjukkan dari mana data mengalir dan kemana data akan menuju, keterangan arus data ini perlu dicatat di dalam kamus data sehingga dapat memudahkan mencari arus data.

5. Penjelasan

Penjelasan digunakan untuk lebih memperjelas lagi tentang makna dari arus data yang dicatat di kamus data, maka bagian penjelasan dapat diisi dengan keterangan-keterangan tentang arus data tersebut.

6. Periode

Periode ini menunjukkan kapan terjadinya arus data, periode perlu dicatat karena dapat digunakan untuk mengidentifikasi kapan *input* data harus dimasukkan ke dalam sistem.

7. Volume

Volume yang perlu dicatat dalam kode adalah tentang volume rata-rata dan volume puncak dari arus data. Volume rata-rata menunjukkan banyaknya rata-rata arus data yang mengalir dalam satu periode tertentu

dan volume puncak menunjukkan volume yang terbanyak. Volume ini digunakan untuk mengidentifikasi besarnya simpanan luar yang akan digunakan, kapasitas, dan jumlah dari alat *input*, alat pemrosesan, dan alat *output*.

8. Struktur Data

Struktur data menunjukkan arus data yang dicatat pada kode yang terdiri dari item-item data apa saja.

2.8. ***Hierarchy plus Input-Process-Output (HIPO)***

Menurut Jogiyanto (2005), HIPO merupakan alat dokumentasi program, akan tetapi sekarang HIPO banyak digunakan sebagai alat desain dan teknik dokumentasi dalam siklus pengembangan sistem. HIPO berbasis pada fungsi, yaitu setiap modul di dalam sistem digambarkan oleh fungsi utamanya.

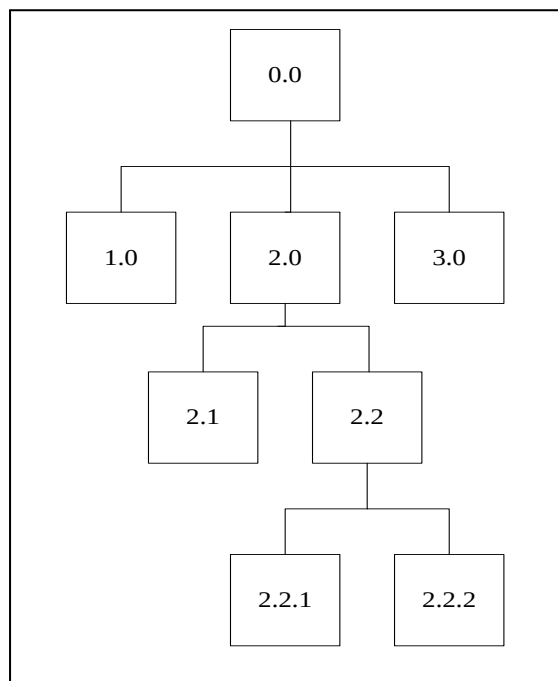
Adapun tujuan dari pembuatan HIPO antara lain:

1. Untuk menyediakan suatu struktur guna memahami fungsi-fungsi dari sistem.
2. Untuk lebih menekankan fungsi-fungsi yang harus diselesaikan oleh program, bukannya menunjukkan pernyataan-pernyataan program yang digunakan untuk melaksanakan fungsi tersebut.
3. Untuk menyediakan penjelasan yang jelas dari *input* yang harus digunakan dan *output* yang harus dihasilkan oleh masing-masing fungsi pada tiap-tiap tingkatan dari diagram-diagram HIPO.
4. Untuk menyediakan *output* yang tepat dan sesuai dengan kebutuhan-kebutuhan pemakai.

HIPO dapat digunakan sebagai alat pengembangan sistem dan teknik dokumentasi program, fungsi-fungsi dari sistem digambarkan oleh HIPO dalam tiga tingkatan. Untuk masing-masing tingkatan digambarkan dalam bentuk diagram tersendiri, dengan demikian HIPO menggunakan tiga macam diagram untuk masing-masing tingkatannya antara lain:

1. *Visual Table Of Contents (VTOC)*

Visual table of contents menggambarkan hubungan fungsi-fungsi di sistem secara berjenjang, VTOC menggambarkan seluruh program HIPO baik rinci maupun ringkasan yang terstruktur. Pada diagram ini nama dan nomor dari program HIPO diidentifikasi. Struktur paket diagram dan hubungan fungsi juga diidentifikasi dalam bentuk hirarki. Keterangan masing-masing fungsi diberikan pada bagian penjelasan yang diikutsertakan dalam diagram ini. Contoh *visual table of contents* dapat dilihat pada Gambar II.7.



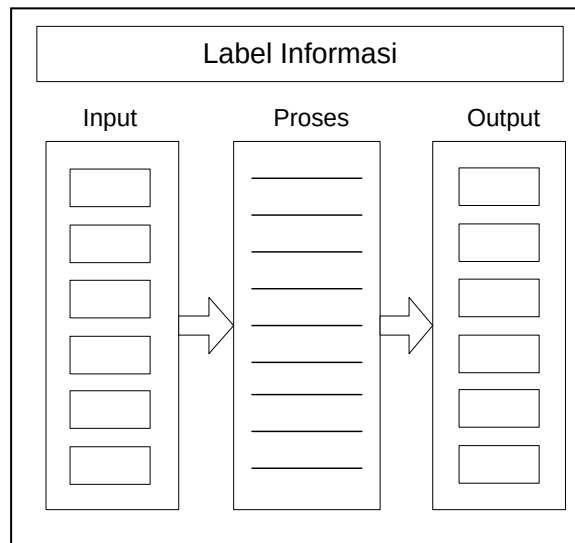
Gambar II.7 Contoh *Visual Table Of Contents*

(Sumber: Jogiyanto, 2005)

2. Overview Diagram

Overview Diagram menunjukan secara garis besar hubungan dari *input*, *proses* dan *output*. Bagian *input* menunjukan *item-item* data yang akan digunakan oleh bagian proses. Bagian proses berisi sejumlah langkah-langkah yang menggambarkan kerja dari fungsi. Bagian *output* berisi dengan *item-item* data yang dihasilkan atau dimodifikasi oleh langkah-

langkah proses. Contoh *Overview Diagram* ini dapat dilihat pada Gambar II.8.



Gambar II.8 Contoh *Overview Diagram*

(Sumber: Jogiyanto, 2005)

2.9. XAMPP Local Server

XAMPP adalah perangkat lunak yang berfungsi sebagai server yang dapat berdiri sendiri (localhost) yang terdiri atas program Apache HTTP Server, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. XAMPP merupakan web server yang mudah digunakan dan dapat melayani tampilan halaman web yang dinamis, perangkat lunak ini tersedia dalam versi GNU (General Public License) yaitu bebas didapatkan tanpa harus membayar (Wikipedia, 2015).

Ada beberapa bagian XAMPP yang biasa digunakan pada umumnya yaitu sebagai berikut:

1. htdocs, adalah folder tempat meletakkan berkas-berkas yang akan dijalankan, seperti berkas PHP, HTML dan script lainnya.
2. phpMyAdmin, merupakan bagian untuk mengelola basis data MySQL yang ada di komputer.

3. Control Panel, yang berfungsi untuk mengelola layanan (service) XAMPP. Layanan (service) ini seperti menghentikan (stop) layanan, ataupun memulai (start) layanan.

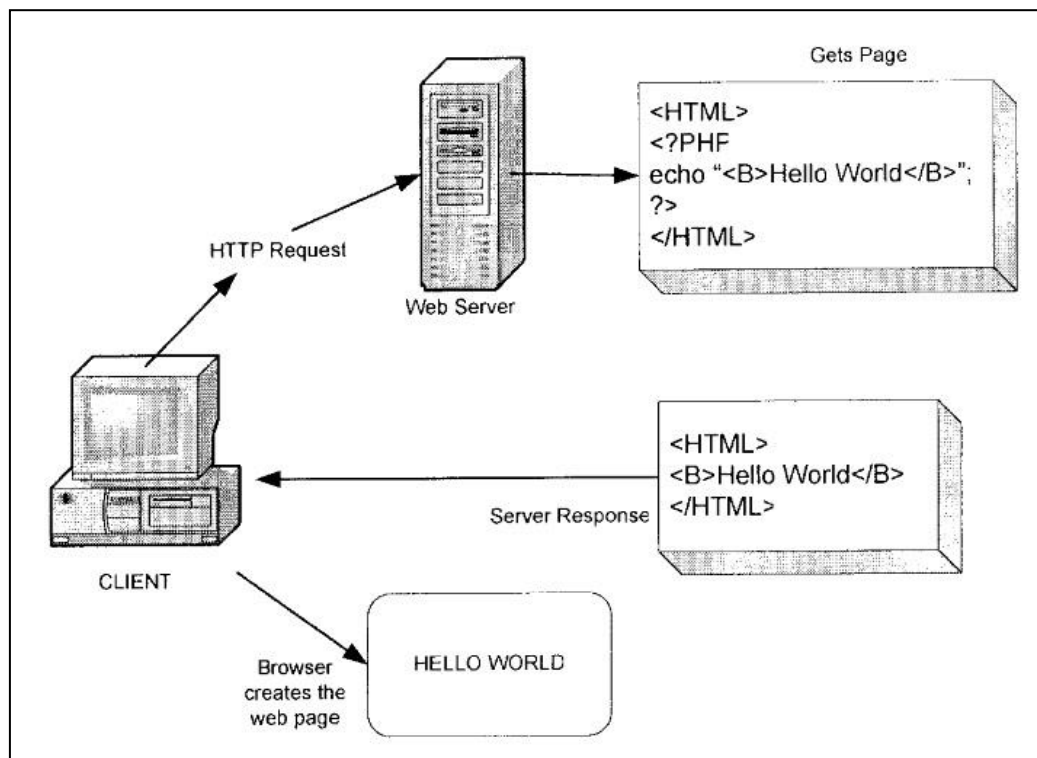
2.10. Hypertext Preprocessor (PHP)

PHP adalah singkatan dari *hypertext preprocessor* yaitu bahasa pemrograman yang memiliki kemampuan untuk memproses dan mengolah data secara dinamis. PHP dapat dikatakan sebagai sebuah *server-side embedded script language* yang artinya semua sintaks dan perintah program yang ditulis akan sepenuhnya dijalankan oleh *server* dan dapat juga diletakkan pada *script hyper text markup language* (HTML).

Pada prinsipnya *server* akan bekerja apabila ada permintaan dari *client*, maka dalam hal ini *client* menggunakan kode-kode PHP untuk mengirim permintaan ke *server* ketika menggunakan PHP sebagai *server-side embedded script language*, maka *server* akan melakukan beberapa hal sebagai berikut (Wahana Komputer, 2010):

1. Membaca permintaan dengan *script* PHP yang berasal dari *browser*.
2. Mencari halaman/*page* di *server* (*server pages*).
3. Melakukan processing melalui instruksi yang diberikan oleh PHP untuk melakukan modifikasi pada halaman/*page*
4. Mengirim kembali halaman tersebut kepada *client* melalui *internet* atau *intranet* yang merupakan proses *echo/print*.

Siklus PHP dalam *web server* dapat dijelaskan pada Gambar II.9.



Gambar II.9 Siklus PHP dalam Web Server

(Sumber: Wahana Komputer, 2010)

2.11. My Structure Query Language (MySQL)

MySQL pertama kali dirintis oleh seorang *programmer database* bernama Michael Widenius. MySQL adalah program *database* yang mampu mengirim dan menerima data dengan sangat cepat dan *multi user*. MySQL memiliki dua bentuk lisensi, yaitu *free software* dan *shareware*. Beberapa tipe data pada MySQL dapat dilihat pada Tabel II.7.

Tabel II.7 Tipe Data pada MySQL

Tipe Data	Ukuran	Keterangan
<i>Char</i>	M	Menampung maksimal M karakter (kombinasi huruf, angka, dan simbol-simbol). Jumlah memori yang dibutuhkan selalu M <i>byte</i> . M terbesar adalah 255.
<i>Varchar</i>	M	Karakter yang disimpan maksimal M karakter.

Tipe Data	Ukuran	Keterangan
		Jumlah memori yang dibutuhkan tergantung jumlah karakter M bisa mencapai 65535.
<i>Date</i>	8 byte	Menyatakan tanggal.

Tabel II.7 Tipe Data pada MySQL (Lanjutan)

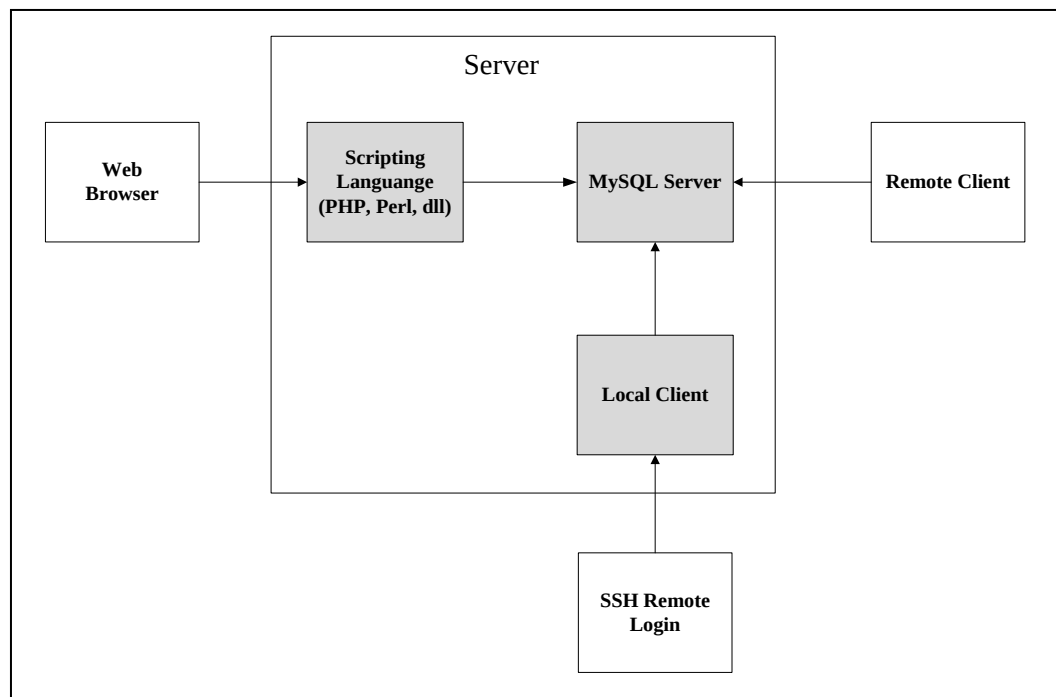
Tipe Data	Ukuran	Keterangan
<i>Time</i>	8 byte	Menyatakan waktu (jam:menit:detik).
<i>Tinyint</i>	1 byte	Bilangan antara -128 sampai dengan +127.
<i>Smallint</i>	2 byte	Bilangan antara -32768 sampai dengan +32768.
<i>Int</i>	4 byte	Bilangan antara -2147683648 sampai dengan +2147683647.
<i>Float</i>	4 byte	Bilangan pecahan.
<i>Double</i>	8 byte	Bilangan pecahan dengan presisi tinggi.
<i>Bool</i>	1 byte	Untuk menampung nilai <i>true</i> (benar) dan <i>false</i> (salah). Identik dengan TINY INT.
<i>Enum</i>	-	Menyatakan suatu tipe yang nilainya tertentu (disebutkan dalam pendefinisian).
<i>Text</i>	-	Menyimpan teks yang ukurannya sangat panjang.
<i>Blob</i>	-	Untuk menyimpan data biner (misalnya gambar atau suara).

(Sumber: Wahana Komputer, 2010)

Keunggulan MySQL adalah kemampuannya dalam menyediakan berbagai fasilitas atau fitur-fitur yang dapat digunakan oleh bermacam-macam *user* seperti

administrator database, programmer aplikasi, manager, sampai dengan end user (pemakai akhir).

Pada mulanya MySQL bekerja pada platform *unix* dan *linux*, namun seiring perkembangannya sekarang banyak bermunculan beberapa versi yang mampu berjalan pada beberapa *platform* yang bersifat *shareware* dan *corporate*. Skema penggunaan MySQL dapat dilihat pada Gambar II.10.



Gambar II.10 Skema *Prototype Server* MySQL
(Sumber: Wahana Komputer, 2010)

MySQL mempunyai prosedur keamanan non-standar yang salah satunya adalah *privillage system* (pengaturan hak akses pengguna), fungsi utama dari sistem ini adalah melakukan pengecekan *user* yang mengakses dari sebuah *host* dan memberikan beberapa hak seperti *SELECT*, *INSERT*, *UPDATE*, dan *DELETE*. MySQL juga mendukung penggunaan koneksi enkripsi yang aman antara *server* dan *client* MySQL menggunakan *secure sockets layer* (SSL) *protocol*. Konfigurasi standar dari MySQL menerapkan kecepatan yang paling tinggi sehingga sambungan enkripsi tidak digunakan secara *default*. Menggunakan

koneksi enkripsi akan menentukan porsi CPU yang besar yang membutuhkan beberapa pekerjaan tambahan sehingga memperlambat kerja MySQL.

MySQL mengizinkan penggunaan enkripsi secara personal, koneksi biasa, atau enkripsi bisa dipilih tergantung dari kebutuhan aplikasi itu sendiri. SSL adalah sebuah protokol yang menggunakan algoritma enkripsi untuk memastikan data yang ditranser melalui *public network* aman. SSL memiliki mekanisme untuk mendeteksi perubahan, kehilangan, atau pengiriman ulang data sehingga SSL memiliki algoritma yang bisa melakukan verifikasi identitas.

BAB III

METODOLOGI PENELITIAN

Metodologi penelitian adalah sekumpulan peraturan, kegiatan dan prosedur yang digunakan dalam melakukan penyelidikan yang sistematis untuk meningkatkan sejumlah pengetahuan, juga merupakan suatu usaha yang sistematis dan terorganisasi untuk menyelidiki masalah tertentu yang memerlukan jawaban oleh pelaku dari suatu disiplin ilmu (Nur dan Bambang, 1999).

3.1. Jenis dan Sumber Data

Sumber data atau informasi penelitian ini berdasarkan kepada jenis data yang diperlukan. Data yang dibutuhkan dalam penelitian ini dikelompokkan menjadi dua jenis, yaitu sumber data primer dan data sekunder.

1. Data primer

Data yang diperoleh melalui pengamatan langsung terhadap sistem yang sedang berjalan dan wawancara dengan pegawai bagian persediaan sebagai sumber informasinya. Dalam penelitian ini data tersebut berupa data alur proses barang masuk dan keluar pada PT Coca Cola Amatil Cibitung Plant.

2. Data sekunder

Data sekunder merupakan sumber data penelitian yang diperoleh secara tidak langsung melalui media perantara, sehingga dapat dikatakan data sekunder merupakan data tambahan dari data primer.

3.2. Metode Pengumpulan Data

Metode pengumpulan data merupakan cara yang dilakukan oleh penulis dalam mengumpulkan data-data yang diperlukan untuk kegiatan penelitian. Berikut metode pengumpulan data yang dilakukan, yaitu:

1. Penelitian Lapangan

a. Observasi

Merupakan kegiatan pengumpulan data dengan cara melakukan peninjauan langsung terhadap sistem yang sedang berjalan yang nantinya akan dianalisis untuk mengetahui permasalahan yang ada. Observasi dilakukan di PT Coca Cola Cibitung Plant pada bagian produksi, gudang, dan pembelian yang berkaitan dengan inventory kontrol.

b. Wawancara

Merupakan kegiatan pengumpulan data dengan cara tanya jawab secara langsung dengan bagian persediaan, gudang, dan pembelian yang berkaitan dengan sistem *inventory*.

2. Studi Pustaka

Dilakukan dengan cara mengumpulkan teori-teori yang berhubungan dengan permasalahan yang akan dibahas.

3.3. Metode Pengembangan Sistem

Prototype adalah versi sistem informasi atau bagian dari sistem yang sudah dapat berfungsi, tetapi dimaksudkan hanya sebagai model awal saja. Setelah beroperasi, *prototype* akan dikembangkan kembali hingga *prototype* tersebut menjadi cocok dengan kebutuhan penggunaannya. Ketika rancangannya telah mencapai tahap final, *prototype* dapat dikonversi menjadi sistem yang lebih baik.

Pembuatan *prototype* meliputi pengembangan sistem uji coba yang cepat dan murah untuk dapat dievaluasi oleh pengguna akhir (*end user*). Lewat interaksi dengan *prototype*, para pengguna dapat memperoleh gagasan yang lebih baik mengenai kebutuhan informasi mereka. *Prototype* yang telah disetujui oleh pengguna dapat digunakan sebagai patokan untuk membuat sistem versi finalnya. *Prototype* terdapat dua jenis yaitu evolusioner dan persyaratan. Untuk penelitian Tugas Akhir ini menggunakan metode pengembangan sistem *evolutionary prototype*. *Prototype* terus menerus disempurnakan sampai memiliki seluruh fungsionalitas yang dibutuhkan pengguna dari sistem yang baru. Jadi,

evolutionary prototype akan menjadi sistem aktual yang dapat dikembangkan lebih lanjut. Dalam perancangan *prototype* pemodelan sistem yang digunakan yaitu *Unified Modelling Language* (UML).

3.4. Kerangka Penelitian

Dalam penelitian yang dibahas dalam tugas akhir ini, dilakukan langkah-langkah atau tahapan dalam penelitian yang ada pada gambar III.1 sesuai dengan metodologi penelitian yang digunakan dalam penelitian tersebut. Penjelasan langkah-langkah atau tahapan dalam penelitian tersebut adalah sebagai berikut:

1. Studi Pendahuluan

Studi pendahuluan yang dilakukan yaitu dengan membaca buku literatur, *browsing* internet serta sumber-sumber lain dalam lingkup perkuliahan maupun di luar lingkup perkuliahan yang berhubungan dengan judul dan permasalahan tugas akhir. Studi pendahuluan ini dimaksudkan untuk dapat mengetahui suatu gambaran yang jelas mengenai kondisi dan situasi PT Coca Cola Amatil Cibitung Plant pada saat ini, serta untuk mengetahui masalah yang sedang dihadapi. Langkah-langkah yang dilakukan dalam studi pendahuluan yang bertujuan untuk mendapatkan informasi adalah dengan cara observasi dan melakukan wawancara terhadap pihak-pihak terkait yang berada di PT Coca Cola Amatil Cibitung Plant.

2. Pokok Permasalahan

Pokok dari permasalahan yang menjadi objek penelitian ini adalah mengenai persoalan pembuatan laporan dan pengecekan barang yang dilakukan masih secara manual menggunakan microsoft excel ,sehingga penyajian laporan kurang informatif.

3. Batasan Masalah

Pembatasan masalah dilakukan agar penelitian yang dilakukan lebih fokus dan lebih terarah.

4. Tujuan Penelitian

Tujuan penelitian dari tugas akhir ini adalah merancang suatu sistem informasi persediaan untuk memudahkan pengawasan/pengendalian persediaan barang di bagian persediaan.

5. Pengumpulan Data

Tahap ini dilakukan untuk mengumpulkan data sebagai bahan yang akan diolah untuk mendeskripsikan hal-hal yang berkaitan dengan sistem pengendalian bahan baku.

6. Pengembangan Sistem Menggunakan Prototipe Evolusioner

Tahap selanjutnya adalah melakukan pengembangan sistem menggunakan metode prototipe evolusioner. Adapun fase-fase yang dilalui pada tahap pengembangan sistem yaitu:

a. Identifikasi Kebutuhan Pengguna

Fase ini dilakukan untuk mengidentifikasi apa saja yang dibutuhkan oleh pengguna pada sistem pengendalian bahan baku.

b. Membuat Prototipe

Membuat sebuah prototipe sesuai dengan analisis yang dilakukan. Pada tahap ini akan dirancang aplikasi tentang sistem pengendalian bahan baku. Perancangan sistem nantinya akan memuat tentang:

1. Pemodelan sistem menggunakan *Unified Modeling Language* (UML).
2. Pemodelan data menggunakan *Class Diagram*.
3. Perancangan sistem dengan menggunakan HIPO, *Flowchart* dan perancangan *interface*.
4. Pembuatan aplikasi dengan menggunakan PHP dan MySQL.

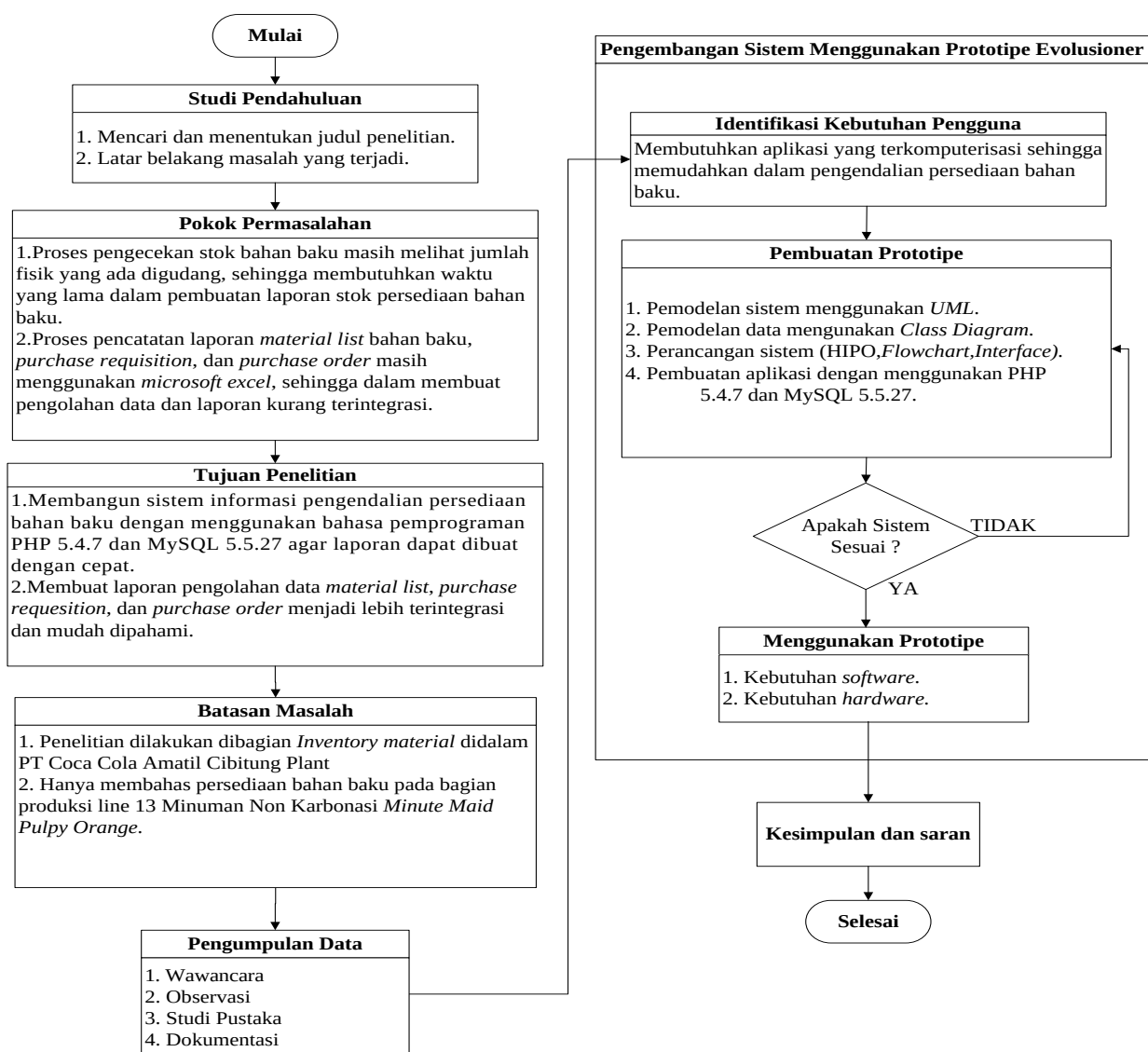
c. Menggunakan prototipe

Memutuskan untuk menggunakan prototipe yang telah divalidasi menggunakan metode prototipe evolusioner, jika tidak sesuai maka akan kembali ke tahap identifikasi.

7. Kesimpulan dan Saran

Mengambil kesimpulan dari hasil analisis sistem berjalan, sistem yang diusulkan dan aplikasi yang dirancang serta memberikan saran kepada PT Coca Cola Amatil Cibitung Plant dan kepada pengembang selanjutnya.

Berikut adalah Gambar III.3 kerangka pemecahan masalah untuk dalam Tugas Akhir ini:



Gambar III.1 Kerangka Pemecahan Masalah

Sumber: Hasil

Pengolahan Data (2015)

BAB IV

PENGUMPULAN DAN PENGOLAHAN DATA

4.1. Pengamatan dan Pengumpulan Data

Pengamatan dilakukan di bagian *inventory material* (persediaan) yang merupakan salah satu bagian di Departemen Produksi PT Coca Cola Amatil Cibitung *Plant*. Pengamatan dilakukan yaitu untuk menganalisis mengenai bagaimana alur prosedur dari sistem informasi pengendalian bahan baku produksi yang sedang berjalan. Pengamatan dilakukan dengan cara sebagai berikut:

1. Mempelajari dokumen organisasi.
2. Mempelajari kegiatan produksi.
3. Survei kegiatan operasi sistem informasi pengendalian bahan baku yang sedang berjalan.

4.1.1 Sejarah Perusahaan

Coca Cola diperkenalkan pertama kali di Indonesia pada tahun 1927. Botol pertama yang terjual di Indonesia merupakan produk yang diimpor dari industri di Belanda dengan nama Koenig. Pada tahun 1932, Coca Cola diproduksi secara lokal oleh *De Netherlands Indische Mineral Water Fabriek Bottle* di Batavia-Indonesia. Setelah vakum pada tahun 1942 karena peristiwa perang dunia, dimulai operasi Coca Cola lagi di Indonesia oleh *Indonesia Bottle Limited* (IBL) pada tahun 1956. Perusahaan nasional ini didirikan oleh TH Ticoalu, yaitu Tatang Nana dan Harry Handoyo. Pada tahun 1971, *Djaja Beverage Amatil Company* memulai produksi setelah masa revolusi dan memperkenalkan *Sprite*. Pada saat itu, pabrik tersebut dapat memproduksi 1000-1500 *case* setiap harinya dan mempekerjakan 25 orang karyawan dengan didukung oleh 3-7 truk distribusi. Pada tahun 1973, Fanta mulai diperkenalkan di Indonesia.

Pada tahun 1977, *Plant Commercial Product Supply* (CPS) mulai tersedia untuk men-*supply* bahan-bahan dasar produk. Pada tahun 1985, Bangun Wenang di Manado selesai dibangun dan memulai produksi Coca Cola untuk pertama kali. Pada tahun 1986, Coca Cola memulai produksi pertamanya dalam kemasan botol

plastik (PET). Sekitar tahun 2000an, Coca Cola mulai memperkenalkan produk baverage non soda sebagai variasi produk dari produk sebelumnya. Pada tahun 2002, Coca Cola Amatil Indonesia memperkenalkan merek *Frestea* dan mengakuisisi brand local air minum dalam kemasan, Ades.

Pada tahun 2008, Coca Cola Amatil Indonesia memperkenalkan produk *Minute Maid Pulpy* dan *Coke Zero* di Indonesia. Pada tahun 2011, Coca Cola Amatil Indonesia memperkenalkan air minum Ades dalam kemasan botol terang berwarna hijau muda. Pada tahun 2012, Coca Cola Amatil Indonesia membuat tempat produksi baru di Cikedokan-Bekasi. Terakhir, pada tahun 2013 Coca Cola Amatil Indonesia memperkenalkan merek *Aquarius* di Indonesia. Hingga saat ini, Coca Cola Amatil Indonesia mempunyai 15 brand yang terdiri dari berbagai jenis minuman. Produk-produk tersebut terbagi dalam empat jenis, yaitu:

1. Produk Minuman Ringan Berkarbonasi, yang meliputi: *Coca Cola*, *Sprite*, *Fanta All Variant*, *Coca Cola Zero*, *Diet Coke*, *Coke Zero*, *A&W Sarsaparilla*, *Schweppes* dan *Sprite Zero*.
2. Produk Minuman Ringan Non-Karbonasi (*Tea*, *Juice*, dan *Isotonik*), yang meliputi : *Minute Maid*, *Frestea*, *Powerade* dan *Aquarius*.
3. Produk Air Mineral dalam Kemasan, yaitu *Ades*.
4. Produk Minuman Berenergi, yaitu *Burn*.

4.1.2. Visi, Misi, dan Tujuan Perusahaan PT Coca Cola Amatil Cibitung Plant

PT Coca Cola Amatil Cibitung *Plant*. Perusahaan yang bergerak dalam bidang pembuatan minuman mempunyai visi dan misi sebagai berikut:

Visi :

Menjadi perusahaan produsen minuman terbaik di Asia Tenggara.

Misi :

Memberikan kesegaran kepada pelanggan dan konsumen kita dengan rasa bangga dan semangat sepanjang hari, setiap hari.

Dengan adanya visi dan misi yang jelas maka perusahaan ini telah memiliki pondasi dan tujuan pasti serta dapat mengurangi faktor ketidakpastian .

4.1.3. Tujuan Perusahaan

Sebagai perusahaan besar PT Coca Cola Amatil Cibitung *Plant* memiliki beberapa tujuan yang mempengaruhi beberapa aspek diantaranya:

1. Sumber Daya Manusia

Sumber daya manusia ini bertujuan untuk Mengembangkan dan memaksimalkan sumber daya yang ada, menghargai prestasi yang telah dihasilkan.

2. Pelanggan

Tingkat kepuasan pelanggan menjadi faktor utama kemajuan perusahaan serta memperluas daya jangkauan perusahaan.

3. Inovasi

Selalu mencari cara yang lebih baik, serta memberikan ide segar yang lebih inovatif.

4. Keunggulan

Senantiasa melakukan pekerjaan yang terbaik dan mengurangi faktor ketidakpastian dalam melakukan pekerjaan.

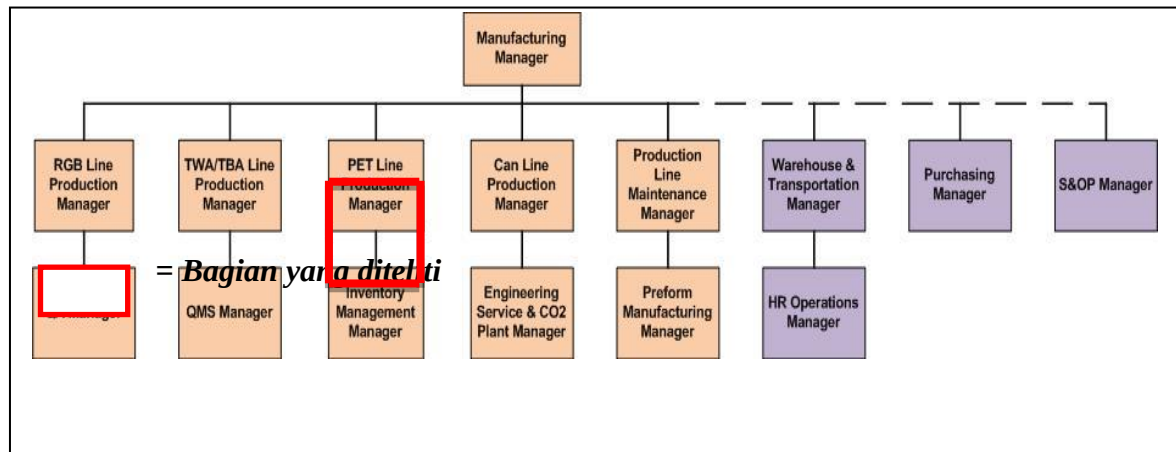
5. Warga Negara yang baik

Melakukan hal yang benar dari perusahaan masyarakat dan sesama kita diharuskan untuk memelihara nilai-nilai perusahaan dengan selalu mempertahankan standar dalam berperilaku.

4.1.4. Struktur Organisasi

PT Coca Cola Amatil Cibitung *Plant* merupakan perusahaan multi nasional yang mempunyai pusat di Atlanta dan Amerika. Coca Cola merupakan salah satu perusahaan minuman terbesar di dunia dan memiliki cabang yang banyak di berbagai negara di seluruh dunia. salah satu yang terdapat di Indonesia terletak di Coca Cola Cibitung Plant Cibitung, Jawa Barat. Semua kendali

operasional perusahaan tetap berada pada organisasi pusat. Untuk struktur organisasi di PT Coca Cola Amatil Indonesia, khususnya Cibitung Plant, kepengurusan tertinggi berada pada *Manufacturing Manager* yang mengatur di Plant. Struktur organisasi PT CCAI Cibitung Plant dapat dilihat pada gambar IV.1.



Gambar IV.1 Struktur Organisasi PT Coca Cola Amatil Cibitung Plant
(Sumber : PT Coca Cola Amatil Cibitung Plant)

Berikut penjelasan fungsi dari setiap bagian yang terdapat pada PT CCAI Cibitung Plant khususnya dalam struktur dimulai dari *Manager Manufacture* ke *Manager Representative* sebagaimana terlihat struktur organisasi dibawah ini:

1. *Manufacturing Manager*

Bertanggung jawab untuk memastikan bahwa Kebijakan Mutu dan *Food Safety* diimplementasikan dan dipelihara, meninjau ketidak sesuaian dalam organisasi dan perencanaan serta melaksanakan tindakan korektif dan pencegahan memverifikasi bahwa solusi telah dilaksanakan dan menyediakan karyawan dengan pekerjaan rutin yang terorganisir dan terstruktur.

2. *Management Representative*

Bertanggung jawab untuk mempertahankan dan memastikan pelaksanaan serta efektifitas dari *Quality Management System* dan *Food Safety*

Management System sesuai dengan standar dari pemerintah, keamanan pangan, dan persyaratan dari Sistem Manajemen yang diadopsi.

3. *Demand Operation Planning Manager (DOP)*

Bertanggung jawab untuk membuat perencanaan produksi, perencanaan material yang dibutuhkan untuk menjamin ketersediaan stok.

4. *Warehouse dan Transportation Manager*

Bertanggung jawab untuk menerima dan menyimpan produk jadi dari produksi, mengelola dan mengontrol pengiriman produk jadi. untuk pusat distribusi dan unit operasi lain sesuai dengan kebutuhan, mengatur lalu lintas truk dan *forklift* selama proses bongkar muat, menjamin penanganan produk yang tepat dan akurasi data.

5. *Preform Manufacturing Manager*

Bertanggung jawab efektifitas operasi dari proses produksi *preform*.

6. *Engineering Services & CO₂ Plant Manager*

Bertanggung jawab untuk menjamin pemeliharaan gedung dalam kondisi yang sesuai.

7. *Inventory Management Manager*

Bertanggung jawab untuk menerima dan menyimpan *RGB (Returnable Glass Bottle)* kosong, peti, palet, suku cadang dan bahan baku untuk memastikan penanganan produk yang tepat dan akurasi data.

8. *RGB(Returnable Glass Bottle) Line Production Manager*

Bertanggung jawab terhadap keefektifan operasi dari proses produksi di semua area *line RGB* (produksi minuman kemasan), mulai dari persiapan bahan baku sampai produk jadi.

9. *PET (Poly Ethylene Terephthalate) Line Production Manager*

Bertanggung jawab terhadap efektifitas operasi dan proses produksi seluruh *PET* baik *CSD (Carbonated Soft Drink)* dan *NCB (Non Carbonated Bottle)*, persiapan bahan baku sampai produksi jadi tertransfer secara baik dari stock holding point, termasuk produksi sirup minuman.

10. *Human Resource Operation Manager*

Bertanggung jawab mengelola dan mengontrol hubungan perseorangan dan mengelola sumber daya manusia.

11. *Procurement Manager*

Bertanggung jawab untuk mendukung ketersediaan semua material (bahan baku, kemasan, suku cadang, dan bahan kimia, dan jasa yang diterima sebagai spesifikasi perpengguna).

12. *Quality Assurance Manager*

Bertanggung jawab untuk menjamin semua *input* dan *output* produksi sesuai dengan standar mutu dan *food safety*.

13. *Production Line Maintenance Manager*

Bertanggung jawab memastikan semua peralatan lini produksi adalah berjalan dalam kondisi optimal, untuk menghasilkan produk dengan mutu dan *food safety* terbaik biaya perawatan yang efektif.

14. *Can Line Production Manager*

Bertanggung jawab terhadap efektifitas operasi dari proses produksi seluruh PET baik CSD dan NCB, persiapan bahan baku sampai produksi jadi tertransfer secara baik dari *stock holding point*, termasuk produksi sirup/minuman.

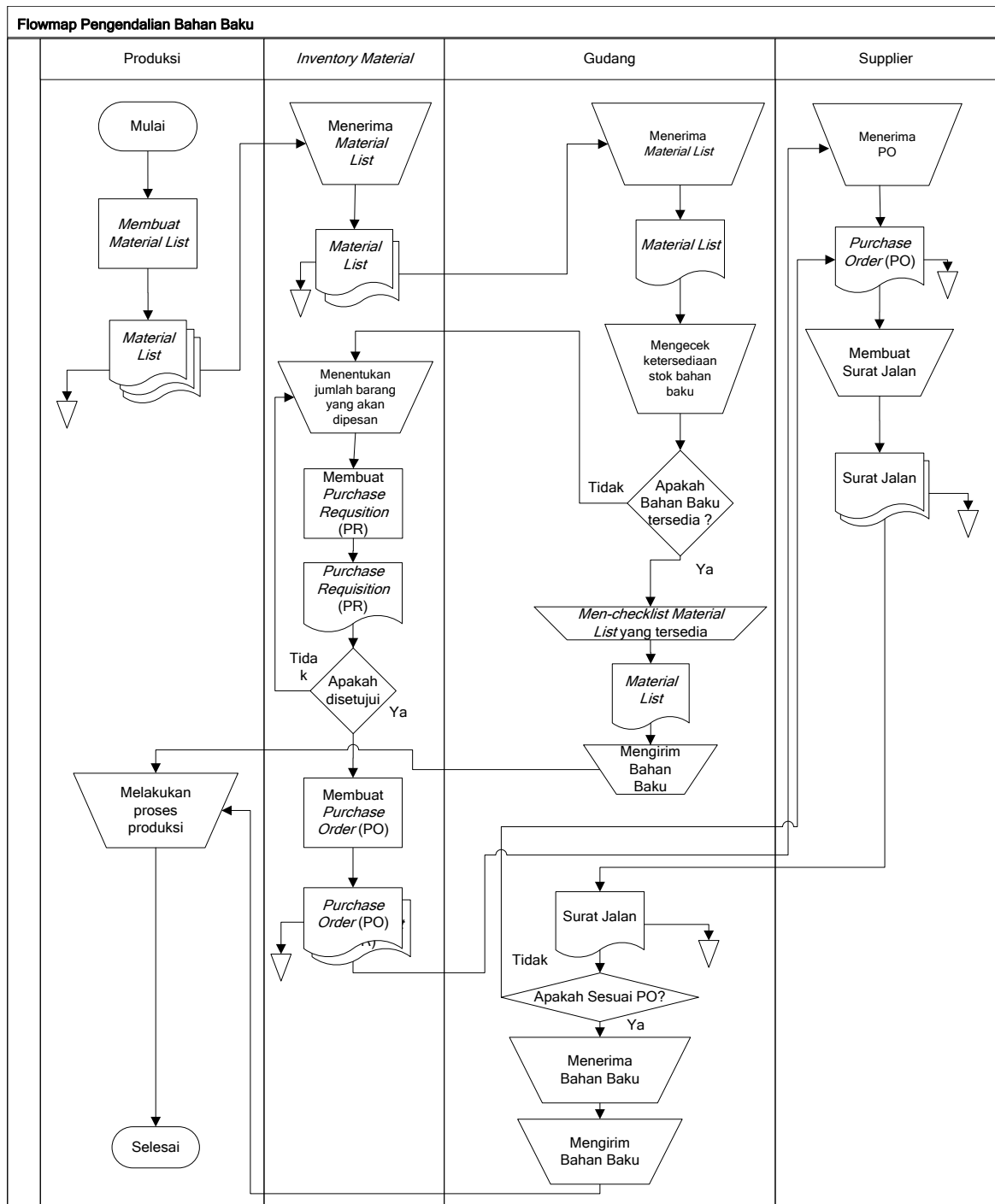
15. *TWA (Tetra Wedge Aseptic)/TWB (Tetra Brik Aseptic) Line Production Manager*

Bertanggung jawab terhadap operasi efektifitas dari proses produksi di area TWA/TWB termasuk *Water Treatment Plant, Waste Water Treatment Plant and Canal Water Treatment and Syrup Manufacturing*.

4.2. Prosedur Sistem Informasi yang Sedang Berjalan

Pada PT Coca Cola Amatil Cibitung *Plant*, bahan baku dikendalikan dengan menghitung stok barang yang ada di gudang dan menggunakan *Microsoft Excel* sebagai media pembuatan dokumen. Hal ini menyebabkan kurang terintegrasinya data sehingga informasi *material* menjadi kurang akurat. Dimulai dari pengendalian stok barang hingga persediaan menipis dan harus dilakukan

pemesanan kembali. Pada kondisi saat ini, perhitungan jumlah pemesanan masih dilakukan dengan hanya melihat perkiraan permintaan produk yang akan dibuat, sehingga terkadang jumlah *material* kurang dalam proses produksi. *Flowmap* sistem pengendalian bahan baku yang sedang berjalan pada PT Coca Cola Amatil Indonesia dapat dilihat pada Gambar IV.2.



Gambar IV.2 *Flowmap* Pengendalian Persediaan Bahan Baku

Pada PT Coca Cola Amatil Indonesia

(Sumber : Hasil Analisis, 2015)

Penjelasan sistem informasi pengendalian bahan baku adalah sebagai berikut:

1. Bagian produksi membuat dokumen *material list* yang diperlukan untuk proses produksi.
2. Bagian *Inventory Material* menerima *material list* dan selanjutnya diberikan ke bagian gudang, kemudian bagian gudang mengecek ketersediaan bahan baku di gudang, apabila bahan baku tersedia maka *material list* akan di *checklist* sesuai dengan stok yang tersedia dan bahan baku dikirimkan ke bagian produksi agar dapat diolah untuk proses produksi.
3. Jika bahan baku tidak tersedia maka bagian *Inventory Material* akan menentukan jumlah barang yang harus dipesan dan membuat *Purchase Requisition (PR)* serta meminta agar *Purchase Requisition (PR)* disetujui *inventory material* agar menjadi *Purchase Order (PO)*.
4. Setelah membuat *Purchase Order*, kemudian PO diberikan kepada *supplier* untuk dibelikan barang yang diminta, kemudian *supplier* memberikan barang yang dipesan kepada bagian gudang untuk diproses kebagian produksi untuk dilakukan proses produksi.

4.2.1. Dokumen yang Terkait Dengan Sistem Informasi Pengendalian Persediaan Bahan Baku yang Berjalan

Dokumen yang terkait dengan sistem informasi pengendalian persediaan bahan baku yang sedang berjalan pada PT Coca Cola Amatil Indonesia adalah sebagai berikut:

1. *Material List*

Berisi rincian material yang akan digunakan untuk proses produksi, *material list* berisikan deskripsi nama material, dan jumlah yang akan dibutuhkan.

Dokumen *material list* dapat dilihat pada Gambar IV.3.

		MATERIAL LIST			
		Material List No.		1009	
		Material List Date		3/31/2015	
		Section		Pulpy Orange	
No.	Material Id	Material Name	Unit	Quantity	Check
TOTAL					
Planner		Head Section		Production Manager	
(.....)		(.....)		(.....)	

Gambar IV.3 Dokumen *Material List*
(Sumber : PT Coca Cola Amatil Cibitung Plant, 2015)

2. *Purchase Requisition* (PR)

Ini merupakan dokumen untuk membuat permintaan pembelian barang, agar pihak supplier dapat melakukan proses pengadaan barang yang diminta. Dokumen *purchase requisition* (PR) dapat dilihat pada Gambar IV.4 .

 PURCHASE REQUISITION PT COCA COLA AMATIL INDONESIA					
To	No				
From	Date				
Please order the following items to be delivery by:					
No.	Material id	Material Name	Unit	Stock	Description
TOTAL					
Requested By		Approved By			
(.....)		(.....)			

Gambar IV.4 Dokumen *Purchase Requisition* (PR)
(Sumber : PT Coca Cola Amatil Cibitung Plant, 2015)

3. *Purchase Order* (PO)

Berisi rincian barang yang harus dipesan terdiri dari diskripsi nama barang, jumlah kuantiti pemesanan, tanggal pengiriman serta persetujuan dari pihak-pihak terkait pemesanan bahan baku pengemas. Dokumen *purchase order* (PO) dapat dilihat pada Gambar IV.5.

 PURCHASE ORDER 						
To	No					
From	Date					
No.	Material ID	Material Name	Unit	Stock	Unit Price	Description
SUB TOTAL						
Planner		Shipper		Driver		
(.....)		(.....)		(.....)		
NOTE						

Distribution Document:
 Original : Data Entry
 Copy 1 : Planner
 Copy 2 : Operational
 Copy 3 : Transporter

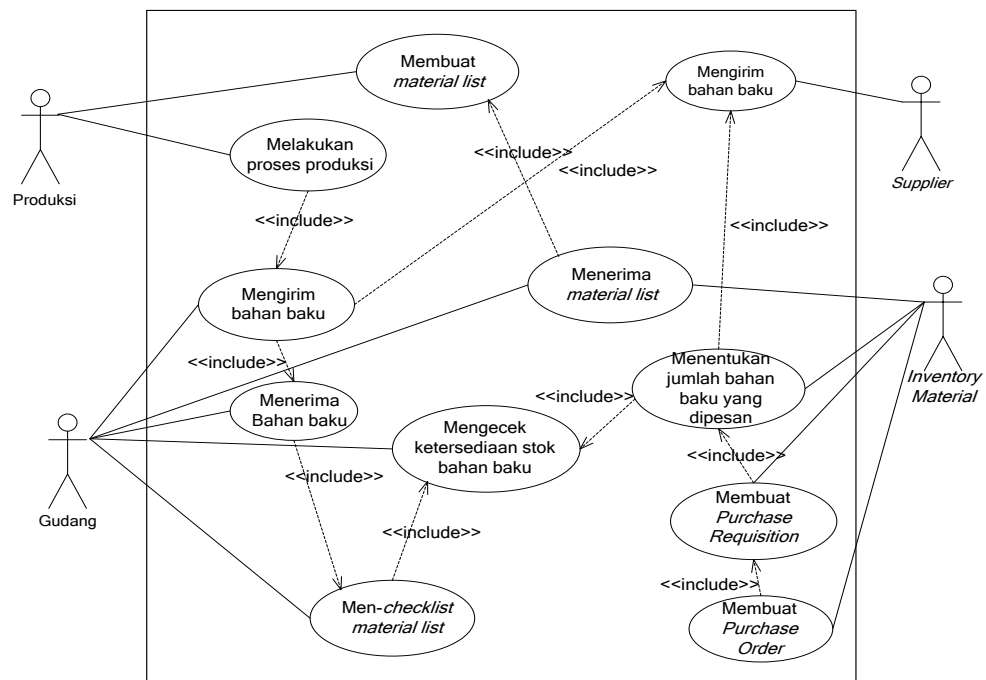
Gambar IV.5 Dokumen *Purchase Order* (PO)
(Sumber : PT Coca Cola Amatil Cibitung Plant, 2015)

4.3 Sistem Informasi Pengendalian Persediaan Bahan Baku Produksi dengan *Unified Modeling Language* (UML)

Penggunaan UML dalam menganalisis sistem informasi pengendalian persediaan bahan baku produksi untuk menyederhanakan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami. Dengan menggunakan UML dapat berkomunikasi dalam perspektif obyek antara *user* dengan *developer*, antara *developer* dengan *developer*, antara *developer* analisis dengan *developer* desain dan antara *developer* desain dengan *developer* pemrograman.

Penggambaran sistem informasi pengendalian persediaan bahan baku dengan menggunakan *use case diagram* bertujuan untuk menggambarkan fungsionalitas dari sebuah sistem yang merepresentasikan sebuah interaksi antara aktor dengan sistem itu sendiri. Dengan menggunakan *use case diagram*, aktivitas yang dilakukan oleh sebuah aktor, dan keterkaitan antara satu aktor dengan aktor yang lain juga.

Sistem informasi pengendalian bahan baku memiliki keterkaitan antar bagian dan aktor diluar sistem, diantaranya adalah bagian produksi, *Inventory Material*, dan bagian gudang. Setiap bagian memiliki kepentingan-kepentingan yang berkaitan dengan pengendalian bahan baku produksi. Pengendalian bahan baku di perusahaan ini berada di bawah kendali bagian *Inventory Material*. *Use case diagram* sistem informasi pengendalian persediaan bahan baku dapat dilihat pada Gambar IV.6.



Gambar IV.6 Use Case Sistem Pengendalian Bahan Baku
 Pada PT Coca Cola Amatil Cibitung Plant
 (Sumber : Pengolahan Data, 2015)

Penggambaran *use case diagram* pada sistem informasi pengendalian persediaan bahan baku pada PT Coca Cola Amatil Cibitung Plant adalah sebagai berikut:

1. Definisi Aktor

Definisi aktor pada sistem informasi pengendalian persediaan bahan baku yang sedang berjalan antara lain:

Tabel IV.1 Definisi Aktor dalam Sistem Informasi Pengendalian Persediaan Bahan Baku

Aktor	Deskripsi
Produksi	Orang yang bertugas untuk melakukan <i>material list</i> dan proses produksi.
Inventory Material	Orang yang bertugas untuk menentukan pembelian bahan baku dan membuat <i>purchase requisition</i> dan <i>purchase order</i>

Gudang	Orang yang bertugas untuk melakukan pengecekan ketersediaan bahan baku dan mengirim bahan baku sampai ke produksi.
<i>Supplier</i>	Orang yang bertugas untuk memberikan material yang dipesan lewat <i>purchase order</i>

(Sumber: Pengolahan Data, 2015)

2. Definisi *Use Case*

Definisi *use case* pada sistem informasi pengendalian persediaan bahan baku yang sedang berjalan antara lain:

Tabel IV.2 Definisi *Use Case* dalam

Sistem Informasi Pengendalian Persediaan Bahan Baku

<i>Use Case</i>	Deskripsi
Membuat <i>Material List</i>	<i>Use case</i> ini menjelaskan bagian produksi yang membuat <i>material list</i> yang diperlukan.
Menerima <i>Material List</i>	<i>Use case</i> ini menjelaskan aktor <i>Inventory Material</i> dan gudang yang menerima <i>material list</i> .
Mengecek Ketersediaan Stok Bahan Baku	<i>Use case</i> ini gudang yang mengecek ketersediaan stok bahan baku.

Tabel IV.2 Definisi *Use Case* dalam

Sistem Informasi Pengendalian Persediaan Bahan Baku (lanjutan)

<i>Use Case</i>	Deskripsi
Men-checklist <i>Material List</i>	<i>Use case</i> ini memungkinkan aktor melakukan <i>checklist</i> terhadap ketersediaan bahan baku apa saja yang tersedia

Mengirim Bahan Baku	<i>Use case</i> ini memungkinkan aktor untuk mengirim bahan baku.
Melakukan Proses Produksi	<i>Use case</i> ini memungkinkan aktor melakukan proses produksi.
Menentukan Jumlah Bahan Baku yang Dipesan	<i>Use case</i> ini memungkinkan aktor menentukan jumlah bahan baku apa saja yang dipesan.
Membuat <i>Purchase Requisition</i>	<i>Use case</i> ini memungkinkan aktor membuat <i>purchase Requisition</i>
Membuat <i>Purchase Order</i>	<i>Use case</i> ini memungkinkan aktor membuat <i>purchase order</i> .
Memberi barang yang dipesan	<i>Use case</i> ini memungkinkan aktor untuk memberi bahan baku sesuai dengan PO.

(Sumber: Pengolahan Data, 2015)

3. Skenario *Use Case*

Definisi *use case* pada sistem informasi pengendalian persediaan bahan baku yang sedang berjalan antara lain:

Tabel IV.3 Skenario *Use Case* Membuat *Material List*

<i>Use Case</i>	<i>Membuat Material List</i>
<i>Actor</i>	Produksi
<i>Brief Description</i>	<i>Use case</i> ini menjelaskan bagian produksi yang membuat <i>material list</i> yang diperlukan.
<i>Normal Flow of Events</i>	Aktor membuat <i>material list</i> yang diperlukan berdasarkan kebutuhan produksi.

(Sumber: Pengolahan Data, 2015)

Tabel IV.4 Skenario *Use Case* Menerima *Material List*

Use Case	Menerima Material List
<i>Actor</i>	<i>Inventory Material, Gudang</i>
<i>Brief Description</i>	<i>Use case ini menjelaskan aktor Inventory Material dan gudang yang menerima material list.</i>
<i>Normal Flow of Events</i>	1. Aktor <i>Inventory Material</i> menerima <i>material list</i> untuk diserahkan kembali ke gudang. 2. Aktor gudang menerima <i>material list</i>.

(Sumber: Pengolahan Data, 2015)

Tabel IV.5 Skenario *Use Case* Mengecek Ketersediaan Stok Bahan Baku

Use Case	Mengecek Ketersediaan Stok Bahan Baku
<i>Actor</i>	<i>Gudang</i>
<i>Brief Description</i>	<i>Use case ini gudang yang mengecek stok bahan baku.</i>
<i>Normal Flow of Events</i>	Aktor mengecek stok bahan baku yang ada di gudang sesuai dengan bahan baku yang diminta.

(Sumber: Pengolahan Data, 2015)

Tabel IV.6 Skenario *Use Case* Men-checklist Material List

Use Case	Menerima Men-checklist Material List
<i>Actor</i>	<i>Gudang</i>
<i>Brief Description</i>	<i>Use case ini memungkinkan aktor melakukan checklist terhadap ketersediaan bahan baku apa saja yang tersedia</i>
<i>Normal Flow of Events</i>	1. Aktor menerima <i>material list</i> dari <i>Inventory Material</i>. 2. Aktor men-checklist bahan baku apa saja yang tersedia di gudang 3. Aktor memberikan kembali <i>material list</i> ke produksi untuk bahan baku tersedia untuk segera melakukan

proses produksi.

(Sumber: Pengolahan Data, 2015)

Tabel IV.7 Skenario *Use Case* Mengirim Bahan Baku

<i>Use Case</i>	Mengirim Bahan Baku
<i>Actor</i>	Gudang
<i>Brief Description</i>	<i>Use case</i> ini memungkinkan aktor untuk mengirim bahan baku.
<i>Normal Flow of Events</i>	Aktor mengirim bahan baku setelah menerima <i>material list</i> mana saja yang akan dikirimkan.

(Sumber: Pengolahan Data, 2015)

Tabel IV.8 Skenario *Use Case* Melakukan Proses Produksi

<i>Use Case</i>	Melakukan Proses Produksi
<i>Actor</i>	Produksi
<i>Brief Description</i>	<i>Use case</i> ini memungkinkan aktor melakukan proses produksi.
<i>Normal Flow of Events</i>	Aktor melakukan proses produksi setelah menerima bahan baku yang diperlukan.

(Sumber: Pengolahan Data, 2015)

Tabel IV.9 Skenario *Use Case* Menentukan Jumlah Bahan Baku yang Dipesan

<i>Use Case</i>	Menentukan Jumlah Bahan Baku yang Dipesan
<i>Actor</i>	<i>Inventory Material</i>
<i>Brief Description</i>	<i>Use case</i> ini memungkinkan aktor menentukan jumlah bahan baku apa saja yang dipesan.
<i>Normal Flow of Events</i>	Aktor menentukan perkiraan jumlah barang yang harus dipesan berdasarkan perhitungan fisik yang ada di gudang

(Sumber: Pengolahan Data, 2015)

Tabel IV.10 Definisi Aktor dan *Use Case* Membuat *Purchase Requisition*

Use Case	Membuat <i>Purchase Requirement</i>
<i>Actor</i>	<i>Inventory Material</i>
<i>Brief Description</i>	<i>Use case ini memungkinkan aktor membuat <i>purchase requirement</i></i>
<i>Normal Flow of Events</i>	Aktor membuat <i>purchase requisition</i> (PR) untuk melakukan pendataan kebutuhan pemesanan.

(Sumber: Pengolahan Data, 2015)

Tabel IV.11 Definisi Aktor dan *Use Case* Membuat *Purchase Order*

Use Case	Membuat <i>Purchase Order</i>
<i>Actor</i>	<i>Inventory Material</i>
<i>Brief Description</i>	<i>Use case ini memungkinkan aktor membuat <i>purchase order</i>.</i>
<i>Normal Flow of Events</i>	Aktor membuat <i>purchase order</i> (PO) berdasarkan keputusan <i>supervisor Inventory Material</i> untuk melakukan pemesanan.

(Sumber: Pengolahan Data, 2015)

Tabel IV.12 Definisi Aktor dan *Use Case* Memberi bahan baku yang dipesan

Use Case	Memberi bahan baku yang dipesan
<i>Actor</i>	<i>Supplier</i>
<i>Brief Description</i>	<i>Use case ini memungkinkan aktor untuk memberi bahan baku sesuai dengan PO.</i>
<i>Normal Flow of Events</i>	Aktor memberi bahan baku yang dipesan sesuai dengan PO yang nantinya akan diberikan untuk bagian gudang.

(Sumber: Pengolahan Data, 2015)

BAB V

ANALISIS DAN PEMBAHASAN

5.1 Analisis Kebutuhan User

Pengadaan bahan baku dengan jumlah yang tepat memberikan banyak keuntungan untuk perusahaan. Keuntungannya adalah meminimalkan biaya simpan dan biaya pesan barang. Pada sistem yang sedang berjalan, pengadaan bahan baku dilakukan berdasarkan keputusan yang diberikan oleh *supervisor Inventory Material* dengan melihat stok fisik yang ada di gudang sehingga bahan baku kurang efektif dan efisien. Berdasarkan hasil pengolahan data dan analisis sistem, dibutuhkan perancangan aplikasi pengendalian bahan baku untuk mengendalikan persediaan dan pengadaan bahan baku pada PT Coca Cola Amatil Cibitung Plant. Perancangan aplikasi ini hanya menampilkan data dan informasi yang berhubungan dengan bahan baku. Perancangan aplikasi yang *user friendly* akan memudahkan *user* dalam menggunakan aplikasi ini.

Kebutuhan program aplikasi dapat dijelaskan melalui proses-proses yang dibutuhkan pemakai dan kebutuhan sistem berdasarkan usulan solusi yang dibuat. Identifikasi kebutuhan sistem komputerisasi menjelaskan proses-proses yang dilakukan oleh program aplikasi, dimana proses-proses tersebut dibutuhkan oleh pemakai. Berikut adalah penjelasan identifikasi kebutuhan pemakai aplikasi pengendalian bahan baku pada PT Coca Cola Amatil Cibitung Plant.

Tabel V.1 Identifikasi Kebutuhan *User*

Kebutuhan Sistem	
Nama Proyek	Aplikasi Pengendalian Persediaan Bahan Baku
Proses Sistem Berjalan	1. Membuat Form Purchase Order (FPO)
Data Masukan (<i>input</i>)	1. Data barang 2. Data <i>supplier</i> 3. Data barang masuk/ keluar

Tabel V.1 Identifikasi Kebutuhan *User* (lanjutan)

Kebutuhan Sistem

Data Keluaran (<i>output</i>)	1. <i>Purchase Requisition</i> 2. <i>Purchase Order</i> 3. Laporan perhitungan EOQ
Model Sistem Informasi	1. Input : Data barang Proses : Pengolahan data barang Output : Laporan barang masuk/ keluar 2. Input : Data barang masuk/ keluar Proses : Pengolahan data barang masuk/ keluar Output : Laporan stok 3. Input : Data barang Proses : Perhitungan EOQ Output : Laporan perhitungan EOQ

(Sumber: Hasil Analisis Data, 2015)

5.2 Analisis Sistem

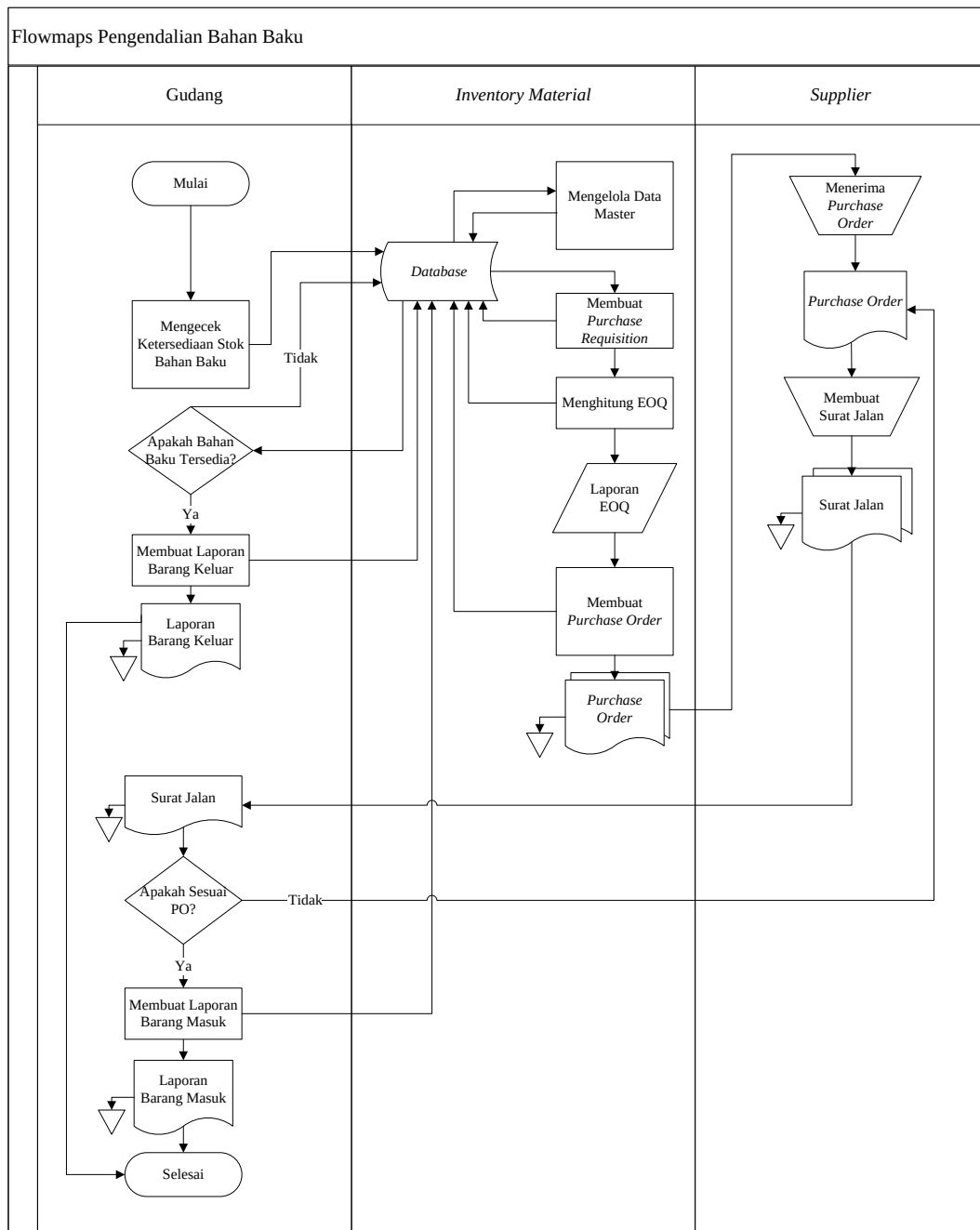
Tujuan perancangan sistem yang diusulkan untuk memberikan penjelasan atau mendefinisikan kepada pemakai program atau *user*. Pembuatan program ini diharapkan dapat membantu mengatasi kekurangan yang ada dan dapat menghasilkan informasi dengan cepat dan tepat.

Tujuan utama dari perancangan sistem adalah untuk memberikan gambaran secara umum kepada pemakai mengenai sistem informasi yang baru. Adapun tujuan dari perancangan sistem yang diusulkan yaitu:

1. Memperbaiki sistem manual dengan sistem komputerisasi agar kinerja kerja lebih efektif dan efisien.
2. Mengelola dan memberikan informasi yang cepat dan akurat kepada *user*.
3. Mempermudah dalam pengambilan keputusan

5.2.1 Flowmap Diagram yang Diusulkan

Flowmap menggambarkan aliran proses yang diusulkan dan aktor yang terlibat. Pada Gambar V.1 dijelaskan *flowmap* usulan aplikasi pengendalian persediaan bahan baku pada PT Coca Cola Amatil Cibitung Plant.

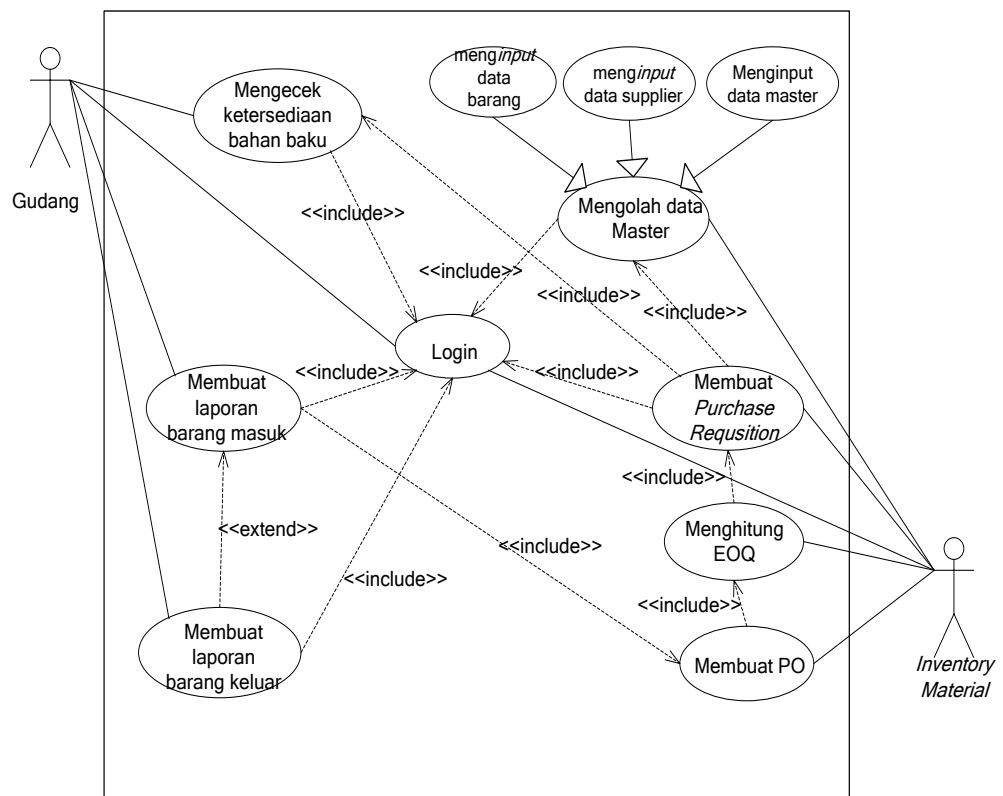


Gambar V.1 FlowMap Diagram yang Diusulkan

(Sumber: Hasil Analisis Data, 2015)

5.2.2 Use Case Diagram yang Diusulkan

Use case diagram menggambarkan fungsionalitas dari sebuah sistem. *Use case diagram* menggambarkan sebuah interaksi antara aktor dengan sistem. Pada *use case diagram* ini terdapat empat aktor yaitu admin, gudang, *Inventory Material*. Berdasarkan analisis yang telah dilakukan, maka *use case diagram* yang diusulkan pada aplikasi pengendalian bahan baku pada PT Coca Cola Amatil Indonesia dapat dilihat pada Gambar V.1 berikut ini:



Gambar V.2 Use Case Diagram yang Diusulkan

(Sumber: Hasil Analisis Data, 2015)

Berikut adalah skenario *use case diagram* proses yang diusulkan:

1. Skenario Use Case Login

Tabel V.2 Skenario Use Case Login

Actor	Semua Aktor
<i>Description</i>	Sistem ini memungkinkan aktor mengakses sistem aplikasi persediaan dengan metode EOQ

Tabel V.2 Skenario *Use Case Login* (lanjutan)

Actor	Semua Aktor
<i>Normal flow</i>	1. Memasukan <i>username</i> 2. Memasukan <i>password</i> 3. Melakukan identifikasi
<i>Alternate flow</i>	Tidak dapat memasuki sistem apabila <i>username</i> dan <i>password</i> tidak terdaftar ataupun melakukan salah pengetikan.

(Sumber: Hasil Analisis Data, 2015)

2. Skenario *Use Case* Mengecek ketersediaan Stok Bahan Baku

Tabel V.4 Skenario *Use Case* Mengecek Ketersediaan Bahan Baku

Actor	Gudang
<i>Description</i>	<i>Use case</i> ini menjelaskan bagian gudang yang mengecek ketersediaan stok bahan baku yang diperlukan untuk proses produksi.
<i>Normal flow</i>	1. Aktor memilih menu <i>material list</i> . 2. Aktor menyesuaikan data yang tersedia.

(Sumber: Hasil Analisis Data, 2015)

3. Skenario *Use Case* Membuat Laporan Barang Keluar

Tabel V.5 Skenario *Use Case* Membuat Laporan Barang Keluar

Actor	Gudang
<i>Description</i>	<i>Use case</i> ini menjelaskan bagian gudang yang membuat laporan barang keluar.

<i>Normal flow</i>	1. Aktor memilih menu barang keluar. 2. Aktor mencari data barang yang akan ditambah berdasarkan id. 3. Aktor dapat mengubah dan menghapus data.
--------------------	--

(Sumber: Hasil Analisis Data, 2015)

4. Skenario *Use Case* Membuat Laporan Barang Masuk

Tabel V.6 Skenario *Use Case* Membuat Laporan Barang Masuk

<i>Actor</i>	Gudang
<i>Description</i>	<i>Use case</i> ini menjelaskan bagian gudang yang membuat laporan barang masuk.
<i>Normal flow</i>	1. Aktor memilih menu barang masuk. 2. Aktor mencari data barang yang akan ditambah berdasarkan id. 3. Aktor dapat menghapus data.
<i>Alternate flow</i>	Tidak dapat menambah data apabila data tidak lengkap.

(Sumber: Hasil Analisis Data, 2015)

5. Skenario *Use Case* Mengolah Data Master

Tabel V.7 Skenario *Use Case* Mengolah Data Master

<i>Actor</i>	<i>Inventory Material</i>
<i>Description</i>	<i>Use case</i> ini menjelaskan bagian <i>inventory material</i> untuk mengolah data master.
<i>Normal flow</i>	1. Aktor memilih submenu data master. 2. Aktor menginput data. 3. Aktor dapat mengubah dan menghapus data.
<i>Alternate flow</i>	Tidak dapat menambah data apabila data tidak lengkap.

(Sumber: Hasil Analisis Data, 2015)

6. Skenario *Use Case* Membuat *Purchase Requisition*

Tabel V.8 Skenario *Use Case* Membuat *Purchase Requisition*

<i>Actor</i>	<i>Inventory Material</i>
<i>Description</i>	<i>Use case ini memungkinkan aktor membuat purchase requisition</i>
<i>Normal flow</i>	1. Aktor memilih menu <i>purchase requisition</i> . 2. Aktor mencari data barang yang akan ditambah berdasarkan id. 3. Aktor dapat menghapus data.

(Sumber: Hasil Analisis Data, 2015)

7. Skenario *Use Case* EOQ

Tabel V.9 Skenario *Use Case* Menghitung EOQ

<i>Actor</i>	<i>Inventory Material</i>
<i>Description</i>	<i>Use case ini memungkinkan Inventory Material untuk menghitung EOQ.</i>
<i>Normal flow</i>	1. Aktor memilih menu hitung EOQ. 2. Aktor mencari data barang yang akan dihitung berdasarkan id. 3. Aktor dapat menyimpan data hasil perhitungan
<i>Alternate flow</i>	Tidak dapat melakukan pencarian data barang apabila id barang tidak sesuai.

(Sumber: Hasil Analisis Data, 2015)

8. Skenario *Use Case* *Purchase Order*

Tabel V.10 Skenario *Use Case* Membuat *Purchase Order*

<i>Actor</i>	<i>Inventory Material</i>
<i>Description</i>	<i>Use case ini memungkinkan aktor membuat purchase order.</i>

<i>Normal flow</i>	Aktor membuat <i>purchase order</i> (PO) berdasarkan dari <i>purchase requisition</i> (PR) yang telah dihitung dari dengan metode EOQ.
--------------------	--

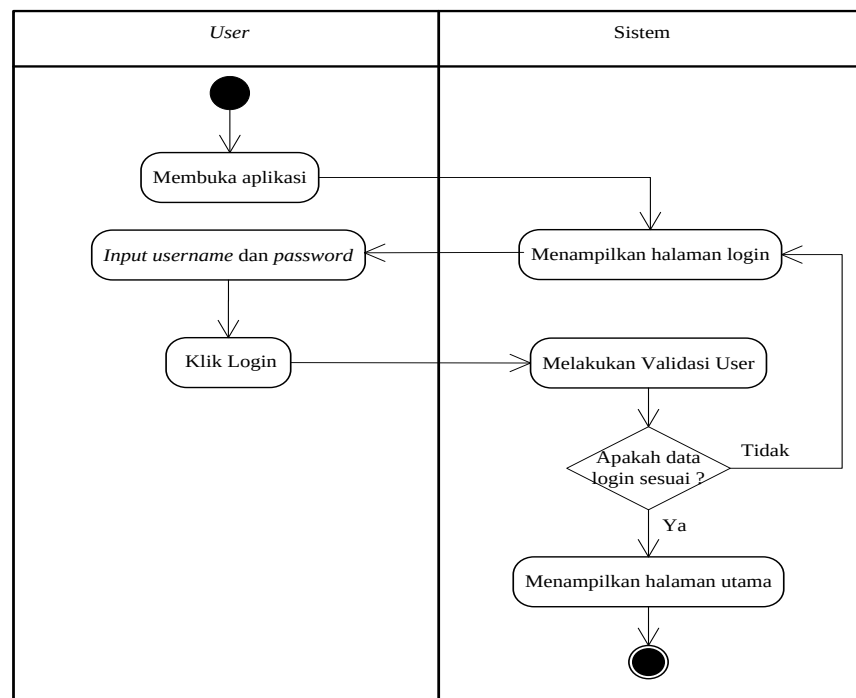
(Sumber: Hasil Analisis Data, 2015)

5.2.3 Activity Diagram yang Diusulkan

Activity diagram digunakan untuk menggambarkan kegiatan-kegiatan yang ada di dalam sistem. Agar lebih memahami sistem yang akan dibuat, maka perlu dibuatkan *activity diagram* untuk aplikasi pengendalian bahan baku. Berikut adalah *activity diagram* yang diusulkan :

1. Activity Diagram Proses Login

Activity diagram login menjelaskan kegiatan proses *login user* terhadap sistem. *Login* merupakan pintu utama untuk pengaksesan aplikasi pengendalian bahan baku dengan metode EOQ oleh *user*. Terdapat tiga *user* yang dapat mengakses aplikasi ini, masing-masing memiliki hak akses yang berbeda sesuai dengan kebutuhan pada bidang pekerjaannya. Berikut pada Gambar V.3 merupakan *activity diagram* yang diusulkan:

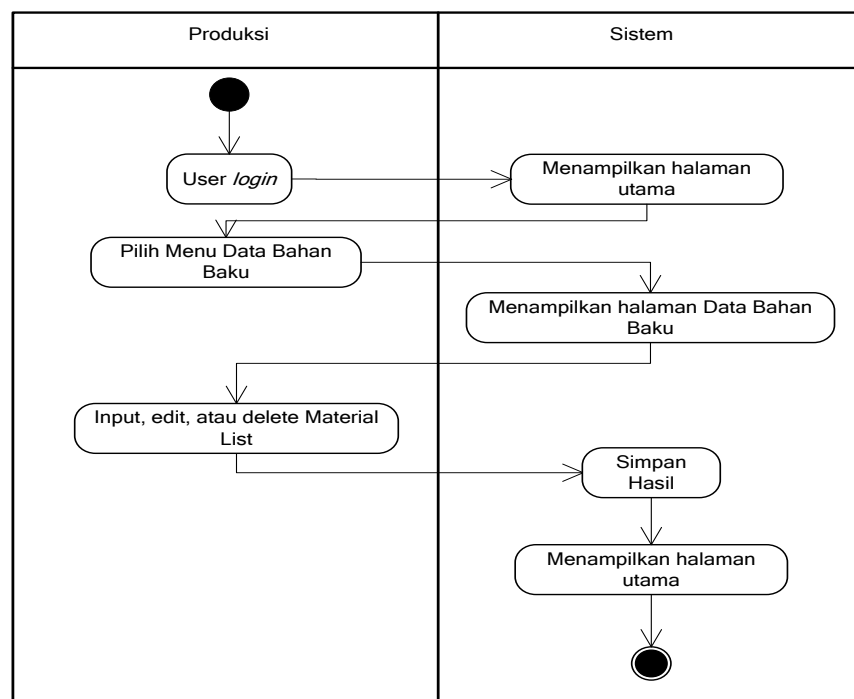


Gambar V.3 Activity Diagram Proses Login yang Diusulkan

(Sumber : Hasil Analisis Data, 2015)

2. *Activity Diagram Membuat Material List*

Activity diagram Membuat Material List menjelaskan kegiatan proses *user* dalam membuat *material list* terhadap sistem. Membuat *Material List* merupakan kegiatan pertama untuk pengendalian bahan baku dengan metode EOQ oleh *user*. Berikut pada Gambar V.4 merupakan *activity diagram* yang diusulkan:

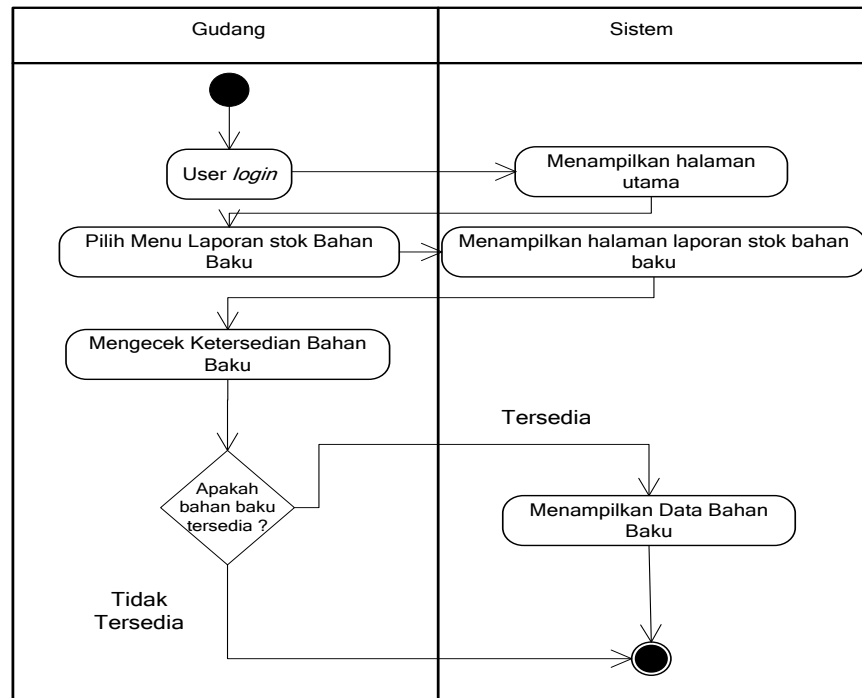


Gambar V.4 *Activity Diagram Membuat Material List*

(Sumber : Hasil Analisis Data, 2015)

3. *Activity Diagram Mengecek Ketersediaan Stok Bahan Baku*

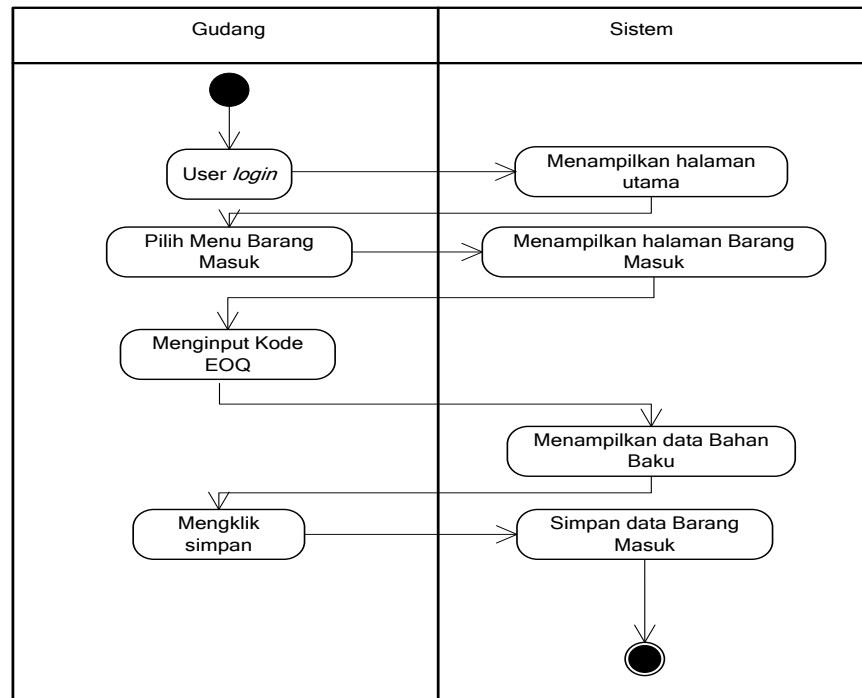
Activity diagram berikut menjelaskan aktivitas untuk mengecek ketersediaan stok bahan baku. *User* yang dapat menjalankan aktivitas ini hanya bagian Gudang. Berikut pada Gambar V.5 merupakan *activity diagram* mengecek ketersediaan stok bahan baku yang diusulkan:



Gambar V.5 Activity Diagram Mengecek Ketersediaan Stok Bahan Baku
(Sumber : Hasil Analisis Data, 2015)

4. Activity Diagram Menerima Bahan Baku

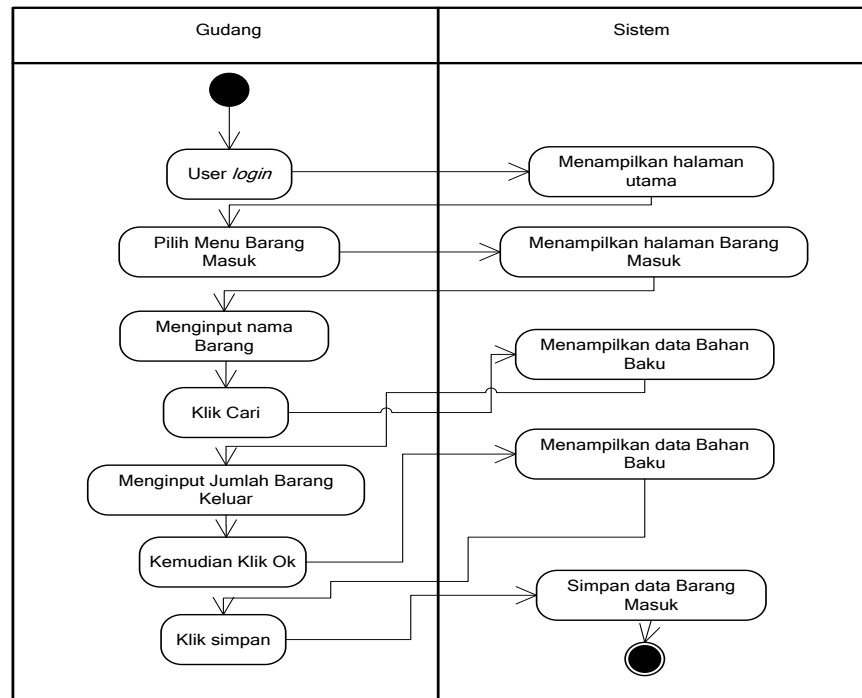
Activity diagram berikut menjelaskan aktivitas untuk menerima stok bahan baku. User yang dapat menjalankan aktivitas ini hanya bagian Gudang. Berikut pada Gambar V.6 merupakan activity diagram menerima stok bahan baku yang diusulkan:



Gambar V.6 *Activity Diagram* Menerima Bahan Baku
(Sumber : Hasil Analisis Data, 2015)

5. *Activity Diagram* Mengirim Bahan Baku

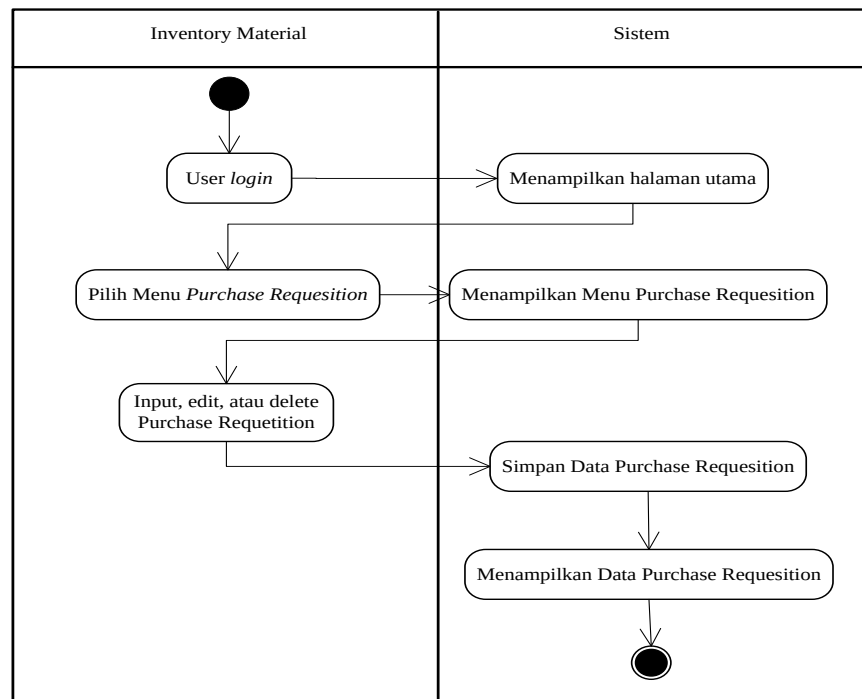
Activity diagram berikut menjelaskan aktivitas untuk mengirim stok bahan baku. *User* yang dapat menjalankan aktivitas ini hanya bagian Gudang. Berikut pada Gambar V.7 merupakan *activity diagram* mengirim stok bahan baku yang diusulkan:



Gambar V.7 Activity Diagram Mengirim Bahan Baku
(Sumber : Hasil Analisis Data, 2015)

6. Activity Diagram Purchase Requisition

Activity diagram berikut menjelaskan aktivitas untuk *Purchase Requisition*. User yang dapat menjalankan aktivitas ini hanya bagian Inventory Material. Berikut pada Gambar V.8 merupakan *activity diagram* purchase requisition yang dipesan yang diusulkan:

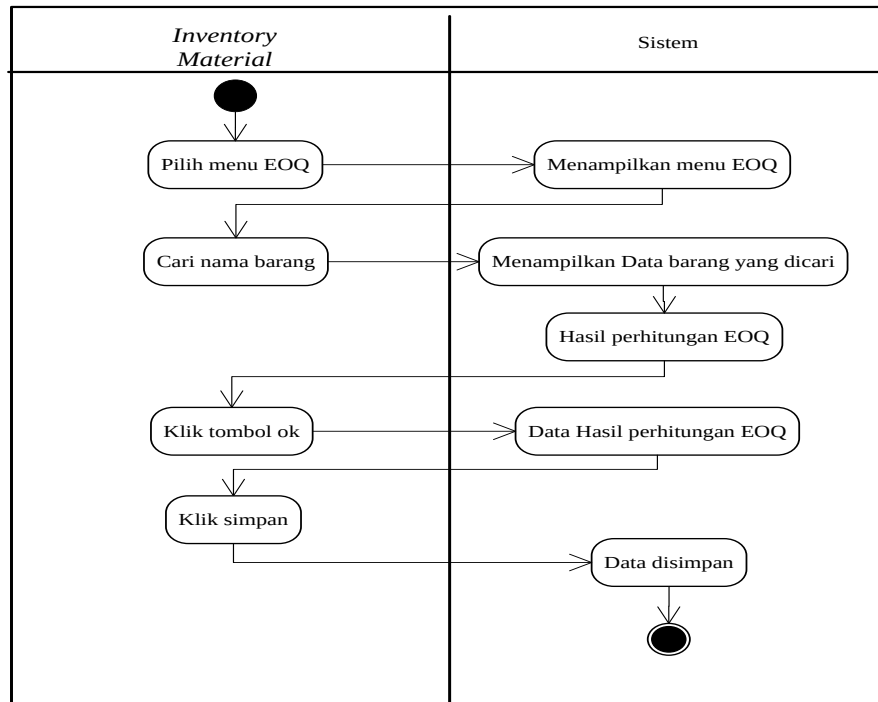


Gambar V.8 Activity Diagram *Purchase Requisition*

(Sumber : Hasil Analisis Data, 2015)

7. Activity Diagram EOQ

Activity diagram berikut menjelaskan aktivitas untuk menghitung EOQ. User yang dapat menjalankan aktivitas ini hanya bagian *Inventory Material*. Proses awal untuk menghitung EOQ adalah dengan mencari data barang sesuai dengan *id* barang. Berikut pada Gambar V.9 merupakan activity diagram EOQ yang diusulkan:

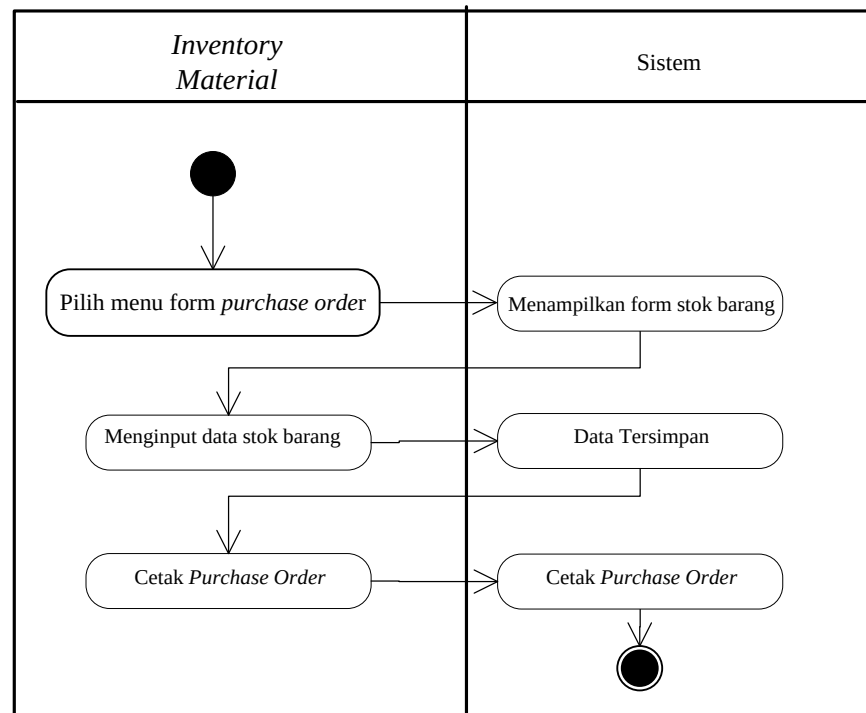


Gambar V.9 Activity Diagram EOQ yang Diusulkan

(Sumber : Hasil Analisis Data, 2015)

8. Activity Diagram Form Purchase Order

Activity diagram berikut menjelaskan aktivitas untuk membuat form purchase order (FPO). User yang dapat menjalankan aktivitas ini hanya bagian *Inventory Material*. Proses awal untuk membuat FPO adalah dengan melihat stok barang yang ada, apabila terdapat stok barang yang telah minimal maka akan dilanjutkan dengan pembuatan FPO. Berikut pada Gambar V.10 merupakan *activity diagram form purchase order* yang diusulkan:



Gambar V.10 Activity Diagram Form Purchase Order yang Diusulkan
(Sumber : Hasil Analisis Data, 2015)

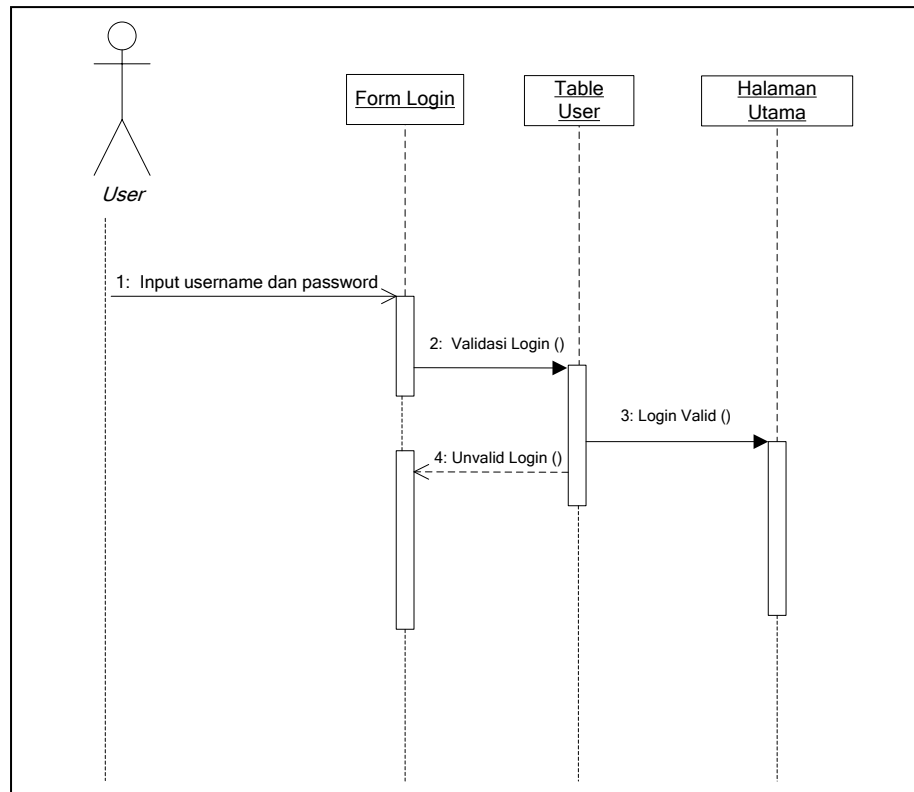
5.2.4 Sequence Diagram yang Diusulkan

Sequence diagram digunakan untuk menggambarkan perilaku pada sebuah *scenario*. Diagram ini menunjukkan sejumlah contoh objek dan *message* yang diletakkan di antara objek-objek di dalam *use case*. *Message* diwakili oleh garis dengan tanda panah dan waktu yang ditujukan dengan *progress vertical*.

1. Sequence Diagram Login

Sequence diagram login user menggambarkan proses yang sedang terjadi dalam *login* admin. Prosesnya dimulai dengan *user* masuk kedalam *form login* terlebih dahulu, kemudian selanjutnya *user* dapat melakukan *login* dengan memasukkan *username* dan *password* untuk dapat masuk kedalam sistem. Proses *login* itu sendiri diikuti juga *validasi* yang dilakukan oleh sistem dimana terdapat proses verifikasi *username* dan *password* apakah sesuai atau tidak. Apabila berhasil *user* tersebut dapat masuk dan

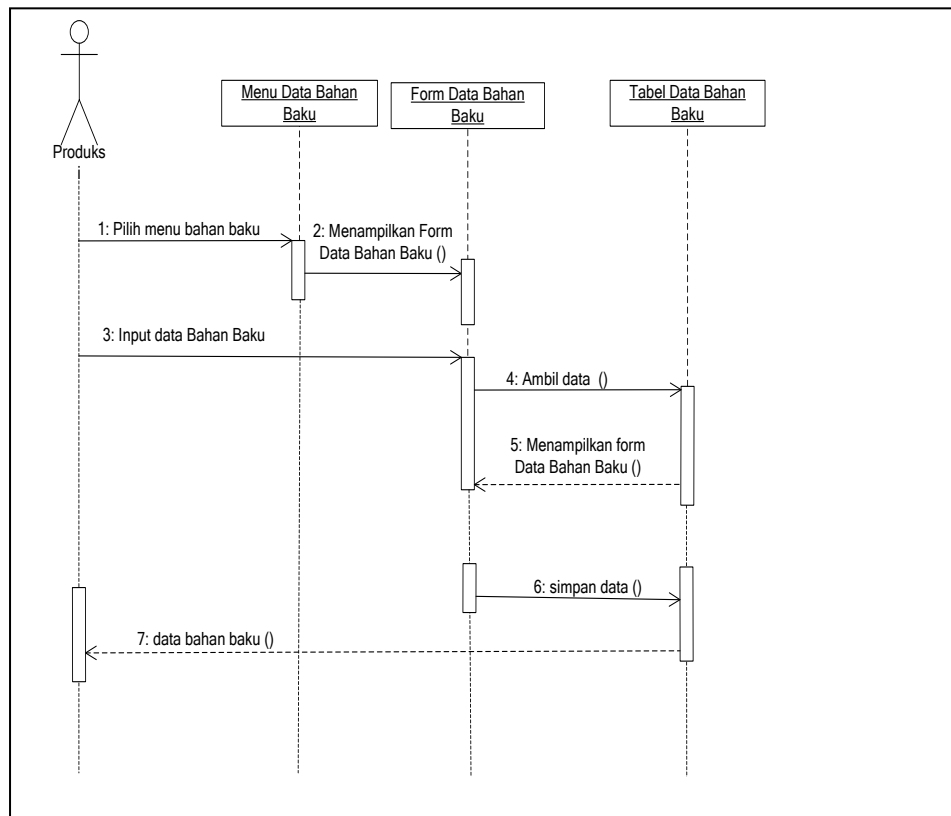
menggunakan sistem, seperti yang dapat dilihat pada Gambar V.11 sebagai berikut:



Gambar V.11 Sequence Diagram Login yang Diusulkan
(Sumber : Hasil Analisis Data, 2015)

2. Sequence Diagram Membuat Material List

Sequence diagram yang merupakan proses membuat *material list*. Diasumsikan bahwa staff produksi sudah melakukan proses *login*, setelah itu staff produksi akan masuk ke halaman utama dan memilih menu data bahan baku untuk membuat *material list*. Seperti yang dapat dilihat pada Gambar V.12 sebagai berikut:

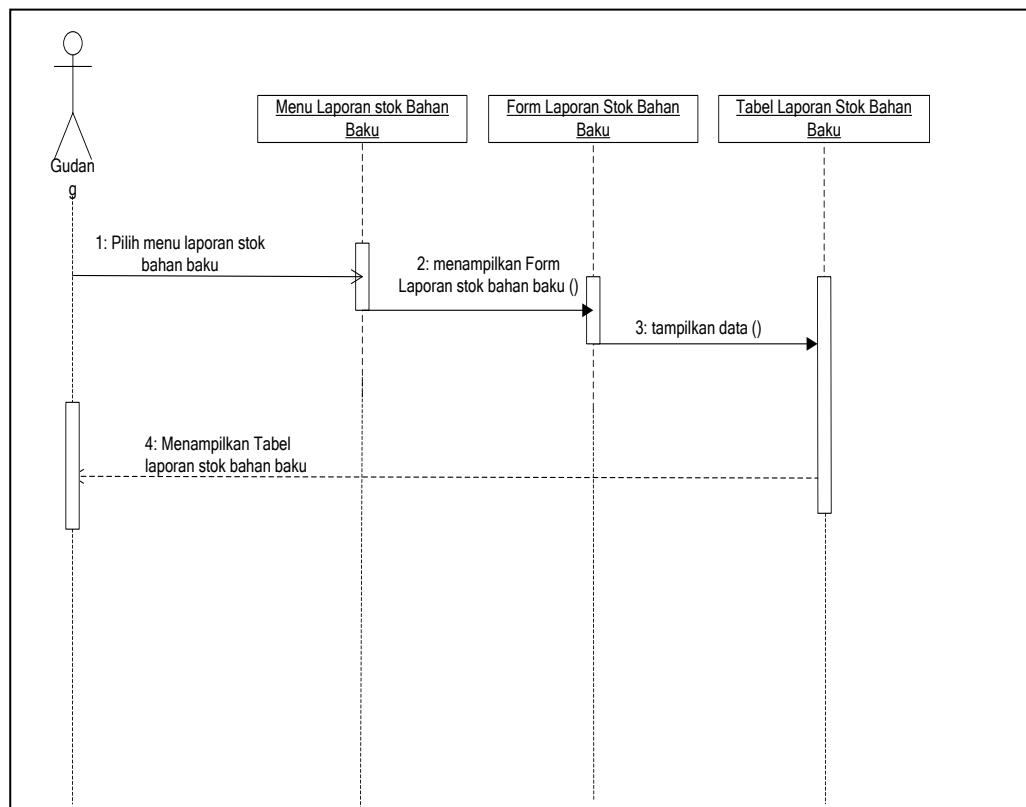


Gambar V.12 *Sequence Diagram* Membuat *Material List*

(Sumber : Hasil Analisis Data, 2015)

3. *Sequence Diagram* Mengecek Ketersediaan Stok Bahan Baku

Sequence diagram yang merupakan proses pengecekan ketersediaan bahan baku. Diasumsikan bahwa gudang sudah melakukan proses *login*, setelah itu gudang akan masuk ke halaman utama dan memilih menu bahan baku untuk mengecek ketersediaan stok bahan baku. Seperti yang dapat dilihat pada Gambar V.13 sebagai berikut:

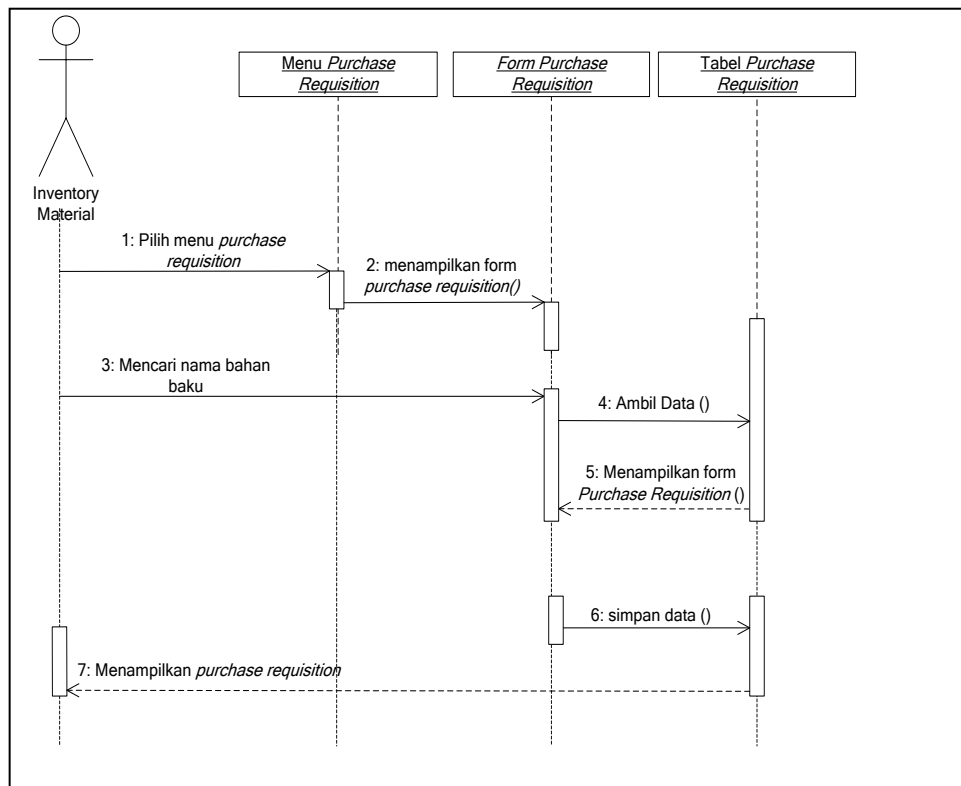


Gambar V.13 *Sequence Diagram* Mengecek Ketersediaan Bahan Baku

(Sumber : Hasil Analisis Data, 2015)

4. *Sequence Diagram* Membuat *Purchase Requisition*

Sequence diagram yang merupakan proses untuk membuat *purchase requisition* dari *inventory material*. Diasumsikan bahwa *purchase requisition* sudah melakukan proses *login*, setelah itu *purchase requisition* akan masuk ke halaman utama dan memilih menu *purchase*. Seperti yang dapat dilihat pada Gambar V.15 sebagai berikut:

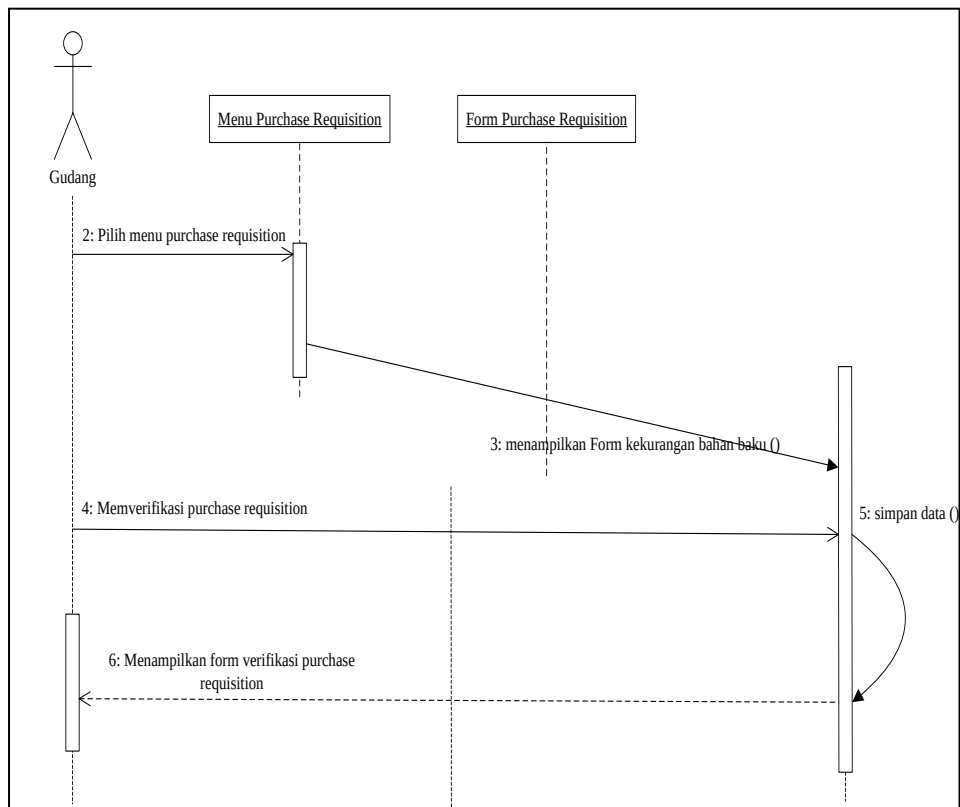


Gambar V.15 *Sequence Diagram* Membuat *Purchase Requisition*

(Sumber : Hasil Analisis Data, 2015)

5. *Sequence Diagram* Verifikasi *Purchase Requisition*

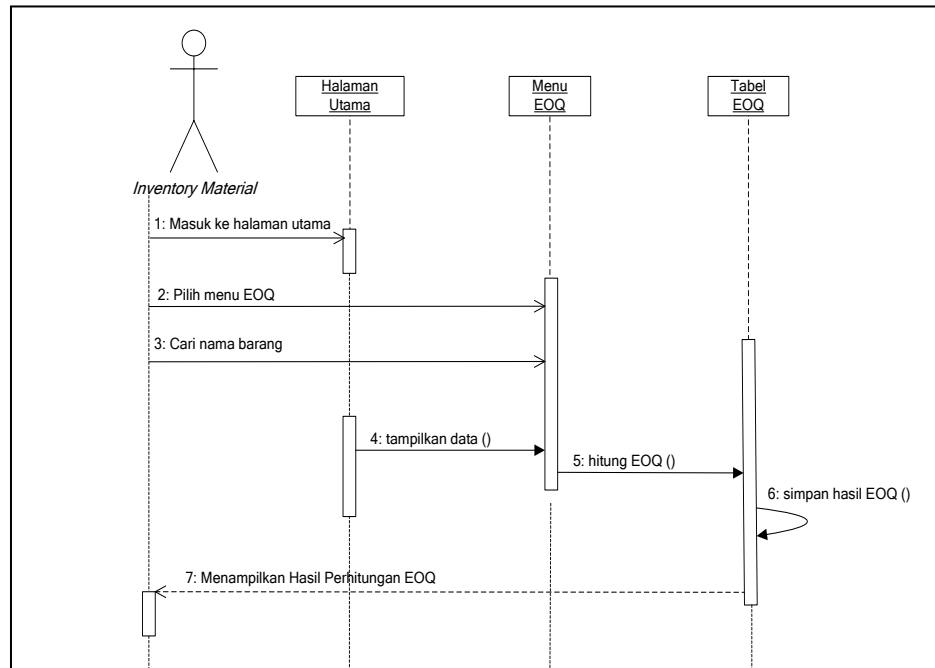
Sequence diagram yang merupakan proses verifikasi *purchase requisition*. Diasumsikan bahwa *inventory material* manajer sudah melakukan proses *login*, setelah itu *inventory material* manajer akan masuk ke halaman utama dan memilih menu *purchase requisition* untuk verifikasi *purchase requisition*. Seperti yang dapat dilihat pada Gambar V.16 sebagai berikut:



Gambar V.16 *Sequence Diagram* verifikasi *purchase requisition*
(Sumber : Hasil Analisis Data, 2015)

6. *Sequence Diagram* EOQ

Sequence diagram yang merupakan proses perhitungan EOQ. Diasumsikan bahwa *Inventory Material* sudah melakukan proses *login*, setelah itu admin akan masuk ke halaman utama dan memilih menu EOQ. Jika *id_barang* tidak sesuai maka proses perhitungan EOQ tidak akan berjalan, seperti yang dapat dilihat pada Gambar V.17 sebagai berikut:

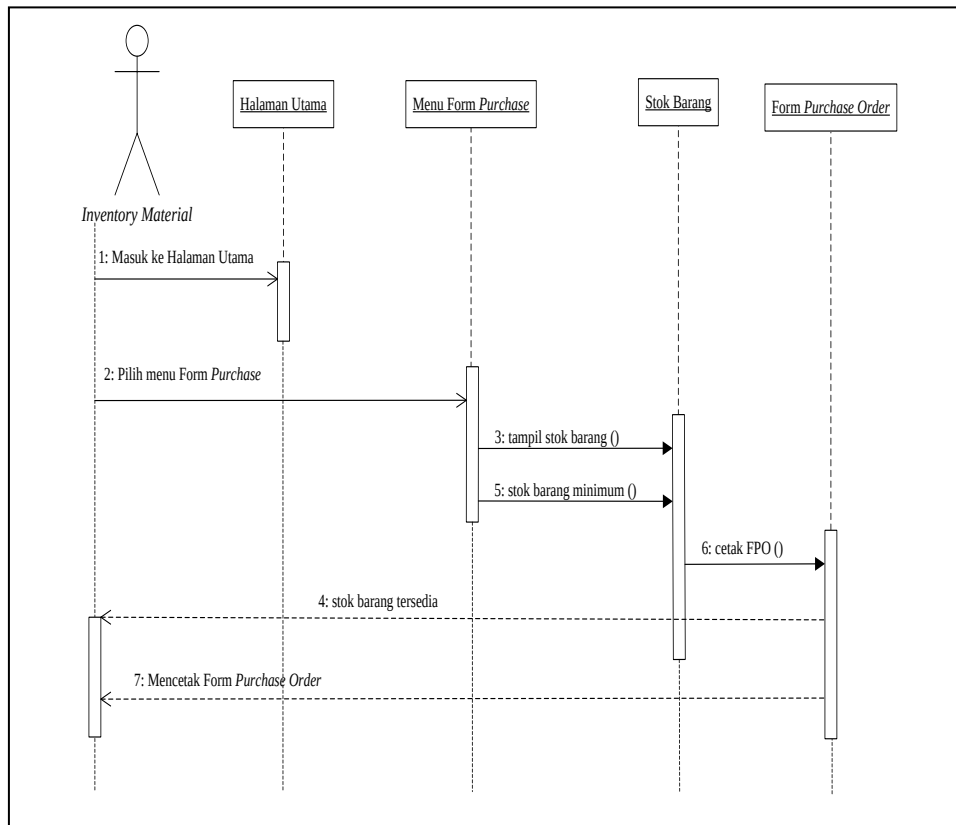


Gambar V.17 *Sequence Diagram* EOQ yang Diusulkan

(Sumber : Hasil Analisis Data, 2015)

7. *Sequence Diagram Form Purchase Order*

Sequence diagram yang merupakan proses untuk membuat form *purchase order* dengan melihat stok barang yang ada. Diasumsikan bahwa *Inventory Material* sudah melakukan proses *login*, setelah itu *Inventory Material* akan masuk ke halaman utama dan memilih form *purchase order*. Apabila stok barang telah minimum, maka akan ada peringatan dan akan dilanjutkan dengan pembuatan form *purchase order*, seperti yang dapat dilihat pada Gambar V.18 sebagai berikut:

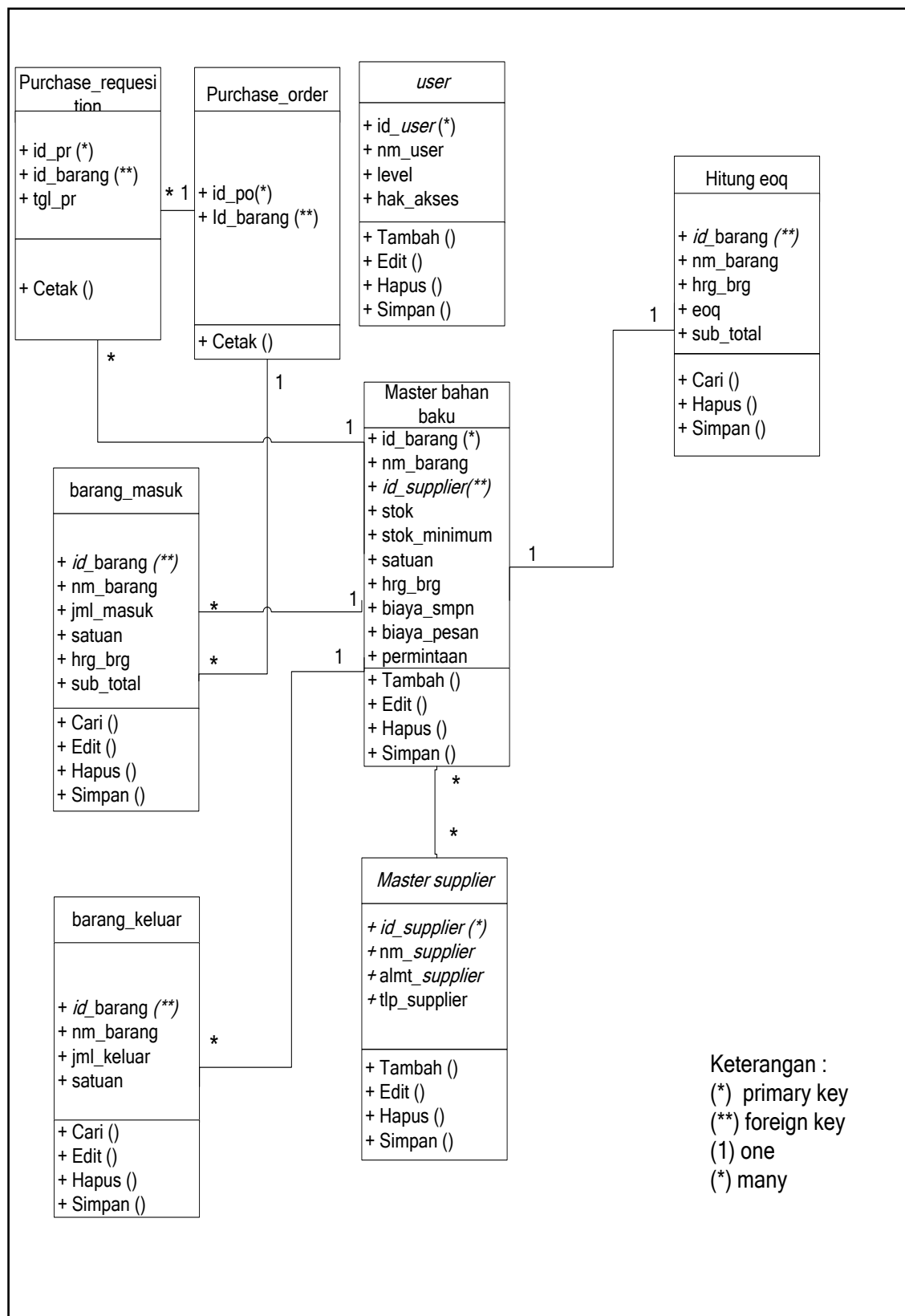


Gambar V.18 *Sequence Diagram* Form Purchase Order yang Diusulkan
(Sumber : Hasil Analisis Data, 2015)

5.2.5 *Class Diagram* yang Diusulkan

Class diagram memperlihatkan hubungan antar kelas dan penjelasan detail tiap-tiap kelas didalam model desain dari suatu sistem. *Class Diagram* membantu dalam visualisasi struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai.

Selama proses analisis, *class diagram* memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. Selama tahap desain, *class diagram* berperan dalam merangkap struktur dari semua kelas yang membentuk arsitektur yang dibuat. Berikut merupakan *class diagram* dalam aplikasi pengendalian bahan baku:



Gambar V.19 Class Diagram Aplikasi Pengendalian Bahan Baku

(Sumber: Hasil Analisis Data, 2015)

5.2.6 Kamus Data

Kamus data adalah suatu daftar data elemen yang terorganisir dengan definisi yang tetap dan sesuai dengan sistem, sehingga *user* dan analis sistem mempunyai pengertian yang sama tentang *input*, *output*, dan komponen *data store*. Pada tahap perancangan sistem, kamus data digunakan untuk merancang *input*, laporan dan basis data. Berikut adalah kamus data pada aplikasi pengendalian bahan baku :

1. Tabel *User*

Tabel V.10 Spesifikasi Tabel *User*

Nama Tabel: *user*

No.	Nama Elemen	Tipe	Panjang	Keterangan
1	id_user	varchar	3	<i>Primary Key</i>
2	nm_user	varchar	15	
3	level	varchar	15	
4	hak_akses	varchar	10	

(Sumber: Hasil Analisis Data, 2015)

2. Tabel *Master Barang*

Tabel V.11 Spesifikasi Tabel *Master Barang*

Nama Tabel: *barang*

No.	Nama Elemen	Tipe	Panjang	Keterangan
1	id_barang	varchar	10	<i>Primary Key</i>
2	nm_barang	varchar	30	
3	id_supplier	varchar	10	<i>foreign key</i>
4	stok	int	5	

5	stok_minimu m	int	5
6	satuan	varcha r	10
7	hrg_brg	int	5
8	biaya_smpn	int	5
9	biaya_pesan	int	5
10	permintaan	int	5

(Sumber: Hasil Analisis Data, 2015)

3. Tabel *Master Supplier*

Tabel V.12 Spesifikasi Tabel *Master Supplier*

Nama Tabel: *supplier*

No.	Nama Elemen	Tipe	Panjan g	Keterangan
1	id_supplier	varchar	10	<i>Primary Key</i>
2	nm_supplier	varchar	30	
3	almt_supplier	varchar	30	
4	tlp_supplier	varchar	15	

(Sumber: Hasil Analisis Data, 2015)

4. Tabel Transaksi Barang Masuk

Tabel V.13 Spesifikasi Tabel Barang Masuk

Nama Tabel: *barang_masuk*

No.	Nama Elemen	Tipe	Panjang	Keterangan
1	id_barang	varchar	10	<i>Foreign Key</i>
2	nm_barang	varchar	10	
3	jml_masuk	int	5	
4	satuan	varchar	10	
5	hrg_barang	int	5	

6	sub_total	int	15
---	-----------	-----	----

(Sumber: Hasil Analisis Data, 2015)

5. Tabel *Purchase Requisition*

Tabel V.14 Spesifikasi Tabel *Purchase Requisition*

Nama Tabel: *purchase_requisition*

No.	Nama Elemen	Tipe	Panjang	Keterangan
1	id_barang	varchar	10	<i>Foreign Key</i>
2	nm_barang	varchar	10	
3	satuan	varchar	10	
4	stok	int	5	

(Sumber: Hasil Analisis Data, 2015)

6. Tabel *Purchase Order*

Tabel V.15 Spesifikasi Tabel *Purchase Order*

Nama Tabel: *purchase_order*

No.	Nama Elemen	Tipe	Panjang	Keterangan
1	id_barang	varchar	10	<i>Foreign Key</i>
2	nm_barang	varchar	10	
3	hrg_beli	int	5	
4	hrg_simpan	int	5	
5	permintaan	int	5	
6	stok	int	5	

(Sumber: Hasil Analisis Data, 2015)

7. Tabel Barang Keluar

Tabel V.16 Spesifikasi Tabel Barang Keluar

Nama Tabel: *barang_keluar*

No.	Nama Elemen	Tipe	Panjang	Keterangan
1	id_barang	varchar	10	<i>Foreign Key</i>
2	nm_barang	varchar	10	
3	jml_keluar	int	5	
4	satuan	varchar	10	

(Sumber: Hasil Analisis Data, 2015)

8. Tabel Hitung EOQ

Tabel V.17 Spesifikasi Tabel Hitung EOQ

Nama Tabel: eoq

No.	Nama Elemen	Tipe	Panjang	Keterangan
1	id_barang	varchar	10	<i>Foreign Key</i>
2	nm_barang	varchar	10	
3	hrng_barang	int	5	
4	eoq	int	5	
5	sub_total	int	15	

(Sumber: Hasil Analisis Data, 2015)

5.2.7 *Deployment Diagram* yang Diusulkan

Deployment diagram menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi seperti dalam gambar V.20. *Deployment diagram* menggambarkan pembuatan aplikasi pengendalian bahan baku berbasis *web* yang memanfaatkan PHP sebagai desain sistemnya. Berikut adalah penjelasan *deployment diagram* aplikasi pengendalian bahan baku:

1. PHP

PHP adalah bahasa pemrograman yang digunakan secara luas untuk penanganan pembuatan dan pengembangan sebuah aplikasi berbasis web dan bisa digunakan bersamaan dengan HTML.

2. *Database*

Database adalah program komputer yang menyediakan layanan data lainnya ke komputer atau program komputer, seperti yang ditetapkan oleh model *client-server*. Istilah ini juga merujuk kepada sebuah komputer yang didedikasikan untuk menjalankan program *server database*.

3. *Web Server*

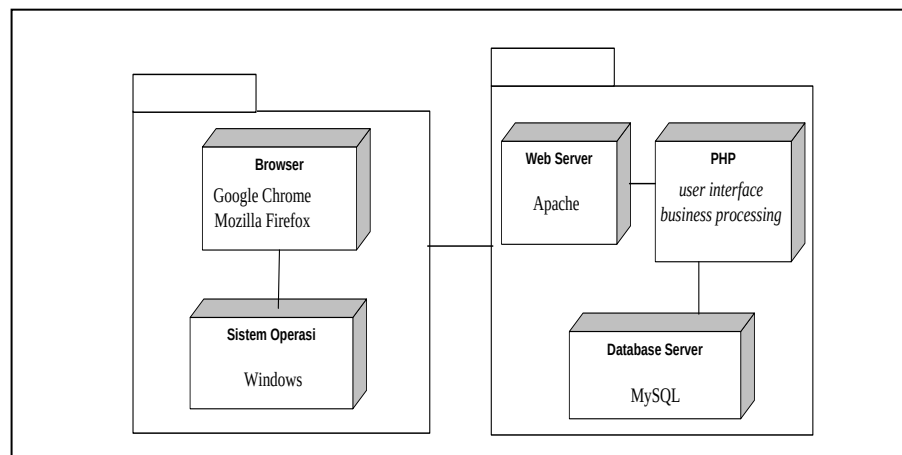
Suatu program sekaligus mesin yang menjalankan program komputer yang mengerti protokol HTTP dan dapat menanggapi permintaan-permintaan dari *web browser*.

4. *Web Browser*

Web browser memiliki arti sebagai penjelajah *web*. Fungsi *web browser* itu sendiri adalah untuk menampilkan dan melakukan interaksi dengan dokumen-dokumen yang disediakan oleh *web server*.

5. Sistem Operasi

Sistem operasi adalah perangkat lunak komputer atau *software* yang bertugas untuk melakukan kontrol dan manajemen perangkat keras dan juga operasi-operasi dasar sistem, termasuk menjalankan *software* aplikasi seperti program-program pengolah data.



Gambar V.20 *Deployment Diagram* yang Diusulkan
(Sumber: Hasil Analisis Data, 2015)

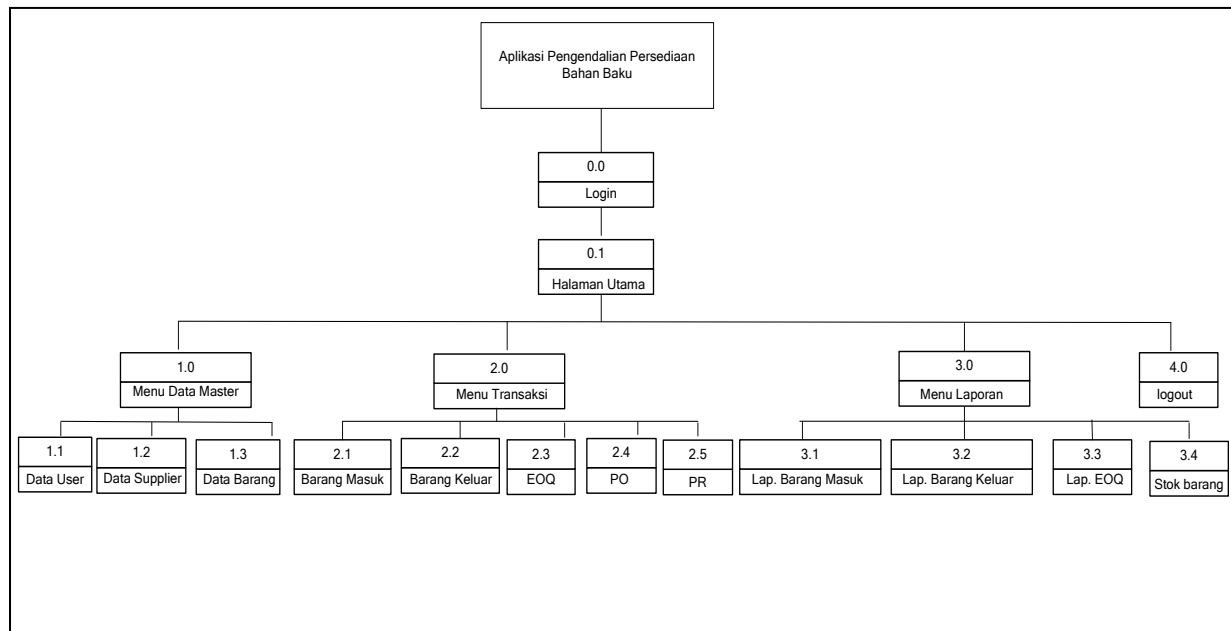
5.3 Perancangan Sistem

Perancangan sistem merupakan suatu kegiatan pengembangan serta perbaikan terhadap sebuah sistem yang berjalan. Pada tahap ini dilakukan upaya untuk memperbaiki sistem ataupun membangun dan menghasilkan sistem yang baru dengan memanfaatkan teknologi terbaru dan fasilitas yang tersedia untuk mengurangi dan mengatasi berbagai permasalahan yang telah terjadi pada sistem yang lama sehingga dapat meningkatkan efektifitas dan efisiensi.

5.3.1 *Hierarchy plus Input-Process-Output (HIPO) Aplikasi*

HIPO merupakan alat desain dan teknik dokumentasi dalam siklus pengembangan sistem. Dalam hal ini pembuatan aplikasi pengendalian bahan baku dibuat menggunakan perangkat lunak PHP untuk bahas pemrograman, serta *MySQL (My Structured Query Language)* yang berfungsi sebagai perangkat lunak basis data.

Berikut adalah struktur menu hirarki program yang digambarkan dalam *HIPO* (*Hierarchy plus Input-Process-Output*):



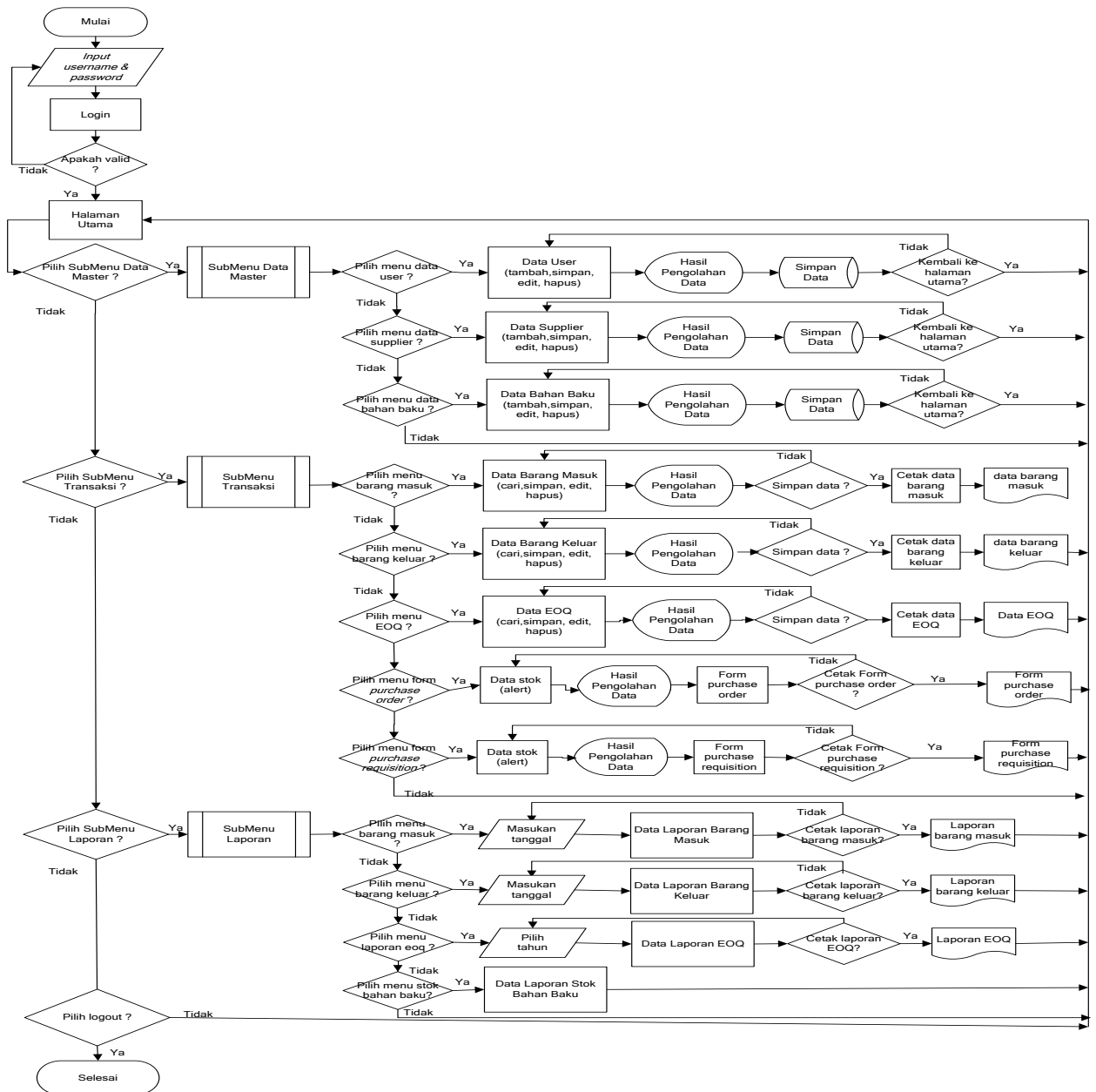
Gambar V.21 HIPO Aplikasi Pengendalian Persediaan Bahan Baku

(Sumber: Hasil Analisis Data, 2015)

5.3.2 *Flowchart* Aplikasi Pengendalian Bahan Baku

Flowchart aplikasi pengendalian bahan baku dapat menggambarkan alur logika yang sebenarnya. Bagian ini juga memperjelas urutan prosedur sistem dan spesifikasi proses. Berikut adalah *flowchart* aplikasi pengendalian bahan baku:





Gambar V.22 Flowchart Aplikasi Pengendalian Bahan Baku
(Sumber: Hasil Analisis Data, 2015)

5.4 Perancangan Input

Perancangan *input* merupakan gambaran *interface* tempat memasukkan data-data kedalam sistem, berikut ini form-form utama untuk *input* data:

1. Form Login

Form *login* digunakan untuk membedakan hak akses pengguna. Melalui *Form login* ini pengguna yang boleh masuk sistem adalah pengguna yang mengetahui *username* dan *password* atau pengguna yang memiliki wewenang untuk menggunakan sistem. Berikut adalah gambar perancangan form login:

LOGIN USER
User Name Password Login

Gambar V.23 Perancangan Form *Login*
(Sumber : Hasil Analisis Data, 2015)

2. Halaman Data *User*

Admin juga dapat menambah, mengubah, serta menghapus data. Berikut adalah gambar perancangan data *user* :

Data Master		Data User											
	Data User	<table border="1" style="margin: auto; border-collapse: collapse;"> <thead> <tr> <th style="padding: 5px;"><i>id_user</i></th> <th style="padding: 5px;"><i>nm_user</i></th> <th style="padding: 5px;">level</th> <th style="padding: 5px;">Hak akses</th> </tr> </thead> <tbody> <tr> <td style="height: 30px;"></td> <td></td> <td></td> <td></td> </tr> </tbody> </table> Tambah Edit Hapus				<i>id_user</i>	<i>nm_user</i>	level	Hak akses				
	<i>id_user</i>					<i>nm_user</i>	level	Hak akses					
Data Bahan Baku													
Data Supplier													
Data Transaksi													
Laporan													
Logout													

Gambar V.24 Perancangan Data *User*
(Sumber : Hasil Analisis Data, 2015)

3. Halaman Data Bahan Baku

Admin juga dapat menambah, mengubah, serta menghapus data. Berikut adalah gambar perancangan data bahan baku:

Data Master		Data Bahan Baku																											
	Data User	<table border="1"> <thead> <tr> <th>ID Barang</th> <th>Nama Barang</th> <th>Supplier</th> <th>Stok</th> <th>Stok min</th> <th>Satuan</th> <th>Harga Barang</th> <th>Biaya Simpan</th> <th>Biaya Pesan</th> <th>Permintaan per tahun</th> </tr> </thead> <tbody> <tr> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> </tr> </tbody> </table> Tambah Edit Hapus								ID Barang	Nama Barang	Supplier	Stok	Stok min	Satuan	Harga Barang	Biaya Simpan	Biaya Pesan	Permintaan per tahun										
	ID Barang									Nama Barang	Supplier	Stok	Stok min	Satuan	Harga Barang	Biaya Simpan	Biaya Pesan	Permintaan per tahun											
	Data Bahan Baku																												
Data Supplier																													
Data Transaksi																													
Laporan																													
Logout																													

Gambar V.25 Perancangan Data Bahan Baku

(Sumber : Hasil Analisis Data, 2015)

4. Halaman Data Supplier

Data supplier akan *diinput* oleh bagian admin. *User* juga dapat menambah, mengubah, serta menghapus data. Berikut adalah gambar perancangan data supplier:

Data Master		Data Supplier											
	Data User	<table border="1"> <thead> <tr> <th>Id Supplier</th> <th>Nama Supplier</th> <th>Alamat</th> <th>Telpn</th> </tr> </thead> <tbody> <tr> <td> </td> <td> </td> <td> </td> <td> </td> </tr> </tbody> </table> Tambah Edit Hapus				Id Supplier	Nama Supplier	Alamat	Telpn				
	Id Supplier					Nama Supplier	Alamat	Telpn					
	Data Bahan Baku												
Data Supplier													
Data Transaksi													
Laporan													
Logout													

Gambar V.26 Perancangan Data Supplier

(Sumber : Hasil Analisis Data, 2015)

5. Halaman Barang Masuk

Halaman ini untuk menginput barang yang masuk ke gudang yang dilakukan hanya oleh bagian gudang dengan mencari id_barang kemudian menginput jumlah yang telah masuk. Berikut adalah gambar perancangan barang masuk:

Data Master		Barang Masuk																	
Data Transaksi		Ketik No Eoq cari																	
	Barang Masuk	Data Barang Masuk Edit Data Hapus Data <table border="1"> <thead> <tr> <th>ID Barang</th> <th>Nama Barang</th> <th>Jumlah</th> <th>Satuan</th> <th>Harga</th> <th>Subtotal</th> </tr> </thead> <tbody> <tr> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> </tr> </tbody> </table>						ID Barang	Nama Barang	Jumlah	Satuan	Harga	Subtotal						
	ID Barang							Nama Barang	Jumlah	Satuan	Harga	Subtotal							
	Barang Keluar																		
	Hitung EOQ																		
Purchase Requisition																			
Purchase Order																			
Laporan		Id Transaksi : Id Supplier Simpan dan cetak Simpan																	
Logout																			

Gambar V.27 Perancangan Barang Masuk
(Sumber : Hasil Analisis Data, 2015)

6. Halaman Barang Keluar

Halaman ini untuk menginput barang yang keluar dari gudang yang dilakukan hanya oleh bagian gudang dengan mencari id_barang kemudian menginput jumlah barang yang telah keluar. Berikut adalah gambar perancangan barang keluar:

Data Master		Barang Keluar																	
Data Transaksi		Catat Barang Keluar Ketik nama barang Cari																	
	Barang Masuk	<table border="1"> <tr> <th>ID Barang</th> <th>Nama Barang</th> <th>Harga Barang</th> <th>Stok</th> <th>Jumlah Keluar</th> <th>Catat</th> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td> ok </td> </tr> </table>						ID Barang	Nama Barang	Harga Barang	Stok	Jumlah Keluar	Catat						ok
	ID Barang	Nama Barang	Harga Barang	Stok	Jumlah Keluar	Catat													
						ok													
	Barang Keluar																		
	Hitung EOQ	Data Barang Keluar Edit Data Hapus Data																	
Purchase Requisition	<table border="1"> <tr> <th>ID Barang</th> <th>Nama Barang</th> <th>Jumlah</th> <th>Satuan</th> </tr> <tr> <td></td> <td></td> <td></td> <td></td> </tr> </table>						ID Barang	Nama Barang	Jumlah	Satuan									
ID Barang	Nama Barang	Jumlah	Satuan																
Purchase Order																			
Laporan																			
Logout		Id Transaksi : Simpan dan cetak Simpan																	

Gambar V.28 Perancangan Barang Keluar
(Sumber : Hasil Analisis Data, 2015)

7. Halaman EOQ

Halaman ini untuk *Inventory Manager* menghitung EOQ dengan hanya mencari id_barang. Berikut adalah gambar perancangan EOQ:

Data Master		Hitung EOQ														
Data Transaksi		HITUNG EOQ Ketik kode PR Cari														
	Barang Masuk	Data Hasil Perhitungan EOQ Hapus														
	Barang Keluar	<table border="1"> <tr> <th>ID Barang</th> <th>Nama Barang</th> <th>Harga Barang</th> <th>EOQ</th> <th>Subtotal</th> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </table>					ID Barang	Nama Barang	Harga Barang	EOQ	Subtotal					
	ID Barang	Nama Barang	Harga Barang	EOQ	Subtotal											
	Hitung EOQ															
Purchase Requisition																
Purchase Order																
Laporan																
Logout		Id Transaksi : Supplier : Simpan														

Gambar V.29 Perancangan EOQ
(Sumber : Hasil Analisis Data, 2015)

8. Halaman Form *Purchase Requisition*

Halaman ini untuk membuat form *purchase requisition* sesuai stok barang yang telah minimum jumlahnya. Berikut adalah gambar perancangan form *purchase requisition*:

Data Master		Purchase Requisition											
Data Transaksi		<table border="1"> <tr> <td>Id Barang</td> <td>Nama Barang</td> <td>Stok</td> <td>Catat</td> </tr> <tr> <td></td> <td></td> <td></td> <td>ok</td> </tr> </table>				Id Barang	Nama Barang	Stok	Catat				ok
Id Barang	Nama Barang	Stok	Catat										
			ok										
	Barang Masuk	Hapus data											
	Barang Keluar	<table border="1"> <tr> <td>Id_barang</td> <td>Nama Barang</td> <td>Satuan</td> <td>Stok</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> </tr> </table>				Id_barang	Nama Barang	Satuan	Stok				
Id_barang	Nama Barang	Satuan	Stok										
	Hitung EOQ												
	Purchase Requisition												
	Purchase Order												
Laporan		Id Transaksi : From : To :											
Logout		Simpan dan Cetak Simpan											

Gambar V.30 Perancangan Form *Purchase Requisition*

(Sumber : Hasil Analisis Data, 2015)

9. Halaman Form *Purchase Order*

Halaman ini untuk membuat form permintaan barang sesuai stok barang yang telah minimum jumlahnya. Berikut adalah gambar perancangan form *purchase order*:

Data Master		Purchase Order																																			
Data Transaksi		Cetak PO																																			
	Barang Masuk	<table border="1"> <thead> <tr> <th>Id_barang</th> <th>Nama Barang</th> <th>Harga Beli</th> <th>Harga Simpan</th> <th>Permintaan Per tahun</th> <th>Stok</th> </tr> </thead> <tbody> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr> </tbody> </table>						Id_barang	Nama Barang	Harga Beli	Harga Simpan	Permintaan Per tahun	Stok																								
	Id_barang							Nama Barang	Harga Beli	Harga Simpan	Permintaan Per tahun	Stok																									
Barang Keluar																																					
Hitung EOQ																																					
Purchase Requisition																																					
Purchase Order																																					
Laporan																																					
Logout																																					

Gambar V.31 Perancangan Form *Purchase Order*

(Sumber : Hasil Analisis Data, 2015)

10. Halaman Laporan Barang Masuk

Halaman ini untuk melihat laporan barang yang telah masuk ke gudang.

User harus menentukan tanggal untuk melihat laporan barang masuk.

Berikut adalah gambar perancangan laporan barang masuk :

Data Master		Laporan Barang Masuk																																							
Data Transaksi		Laporan Barang Masuk Tanggal awal  Tanggal akhir  OK																																							
Laporan																																									
	Laporan Barang Masuk	<table border="1"> <thead> <tr> <th>ID Transaksi</th> <th>Tanggal</th> <th>Supplier</th> <th>Id Barang</th> <th>Nama Barang</th> <th>Jumlah masuk</th> <th>Harga barang</th> <th>Subtotal</th> </tr> </thead> <tbody> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr> </tbody> </table>								ID Transaksi	Tanggal	Supplier	Id Barang	Nama Barang	Jumlah masuk	Harga barang	Subtotal																								
	ID Transaksi									Tanggal	Supplier	Id Barang	Nama Barang	Jumlah masuk	Harga barang	Subtotal																									
Laporan Barang Keluar																																									
Laporan EOQ																																									
Laporan Stok Bahan Baku																																									
Logout																																									

Gambar V.32 Perancangan Laporan Barang Masuk

(Sumber : Hasil Analisis Data, 2015)

11. Halaman Laporan Barang Keluar

Halaman ini untuk melihat laporan barang yang telah keluar dari gudang. *User* harus menentukan tanggal untuk melihat laporan barang keluar. Berikut adalah gambar perancangan laporan barang keluar :

Data Master	Laporan Barang Keluar Laporan Barang Keluar Tanggal awal Calendar Tanggal akhir Calendar OK <table border="1" style="width: 100%;"> <thead> <tr> <th>ID Transaksi</th> <th>Tanggal</th> <th>Id Barang</th> <th>Nama barang</th> <th>Harga Barang</th> <th>Jumlah Keluar</th> <th>Subtotal</th> </tr> </thead> <tbody> <tr> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> </tr> </tbody> </table>	ID Transaksi	Tanggal	Id Barang	Nama barang	Harga Barang	Jumlah Keluar	Subtotal							
ID Transaksi		Tanggal	Id Barang	Nama barang	Harga Barang	Jumlah Keluar	Subtotal								
Data Transaksi															
Laporan															
Laporan Barang Masuk															
Laporan Barang Keluar															
Laporan EOQ															
Laporan Stok Bahan Baku															
Logout															

Gambar V.33 Perancangan Laporan Barang Keluar
(Sumber : Hasil Analisis Data, 2015)

12. Halaman Laporan EOQ

Halaman ini untuk melihat laporan EOQ yang telah dihitung. *User* harus menentukan tanggal untuk melihat laporan EOQ. Berikut adalah gambar perancangan laporan EOQ:

Data Master	Laporan EOQ Laporan EOQ Tahun Calendar OK <table border="1" style="width: 100%;"> <thead> <tr> <th>Id Barang</th> <th>Nama barang</th> <th>Harga barang</th> <th>EOQ</th> </tr> </thead> <tbody> <tr> <td> </td> <td> </td> <td> </td> <td> </td> </tr> </tbody> </table>	Id Barang	Nama barang	Harga barang	EOQ				
Id Barang		Nama barang	Harga barang	EOQ					
Data Transaksi									
Laporan									
Laporan Barang Masuk									
Laporan Barang Keluar									
Laporan EOQ									
Laporan Stok Barang									
Logout									

Gambar V.34 Perancangan Laporan EOQ
(Sumber : Hasil Analisis Data, 2015)

13. Halaman Laporan Stok Bahan Baku

Halaman ini dapat dilihat oleh bagian gudang untuk melihat jumlah stok yang tersedia di gudang. Berikut adalah gambar perancangan laporan stok bahan baku :

Data Master		Laporan Stok Bahan Baku						
Data Transaksi								
Laporan		<u>Laporan Stok Bahan Baku</u>						
	Laporan Barang Masuk	Id Barang	Nama Barang	Harga Beli	Hrg Simpan	Permintaan PerTahun	Stok	Satuan
	Laporan Barang Keluar							
	Laporan EOQ							
	Laporan Stok Bahan Baku							
Logout								

Gambar V.35 Perancangan Laporan Stok Bahan Baku

(Sumber : Hasil Analisis Data, 2015)

5.5 Spesifikasi Perangkat Lunak dan Perangkat Keras

Untuk mendukung kelancaran aplikasi ini selain diperlukan perangkat keras (*hardware*) juga diperlukan perangkat lunak (*software*) yang terdiri dari sistem operasi dan aplikasi *database*. Adapun perangkat lunak yang diperlukan adalah sebagai berikut:

1. Sistem Operasi : *Microsoft Windows XP, Microsoft Windows Vista, Microsoft Windows 7*
2. Web Server : *Apache*
3. Database Server : *MySQL 5.5.27*
4. Design Interface : *PHP 5.4.7*
5. Web Browser : *Mozilla firefox, Google Chrome*

Perangkat keras (*hardware*) adalah suatu perangkat fisik computer yang digunakan untuk memasukkan, memproses, menyimpan dan mengeluarkan hasil

pengolahan data dalam bentuk informasi. Perangkat keras yang dibutuhkan berdasarkan kebutuhan minimal yang harus terpenuhi antara lain:

Tabel V.18 Implementasi Perangkat Keras

Perangkat Keras	<i>Specification</i>
<i>Processor</i>	Intel Core 2 Duo
<i>Main Memory</i>	1 GB
<i>Hard Disk</i>	250 GB
<i>Graphic Adapter (VGA)</i>	500 MB

(Sumber: Hasil Analisis Data, 2015)

BAB VI

KESIMPULAN DAN SARAN

6.1. Kesimpulan

Berdasarkan hasil penelitian selama proses pengumpulan dan pengolahan data, analisis sistem, dan perancangan sistem usulan maka dapat diambil kesimpulan sebagai berikut:

1. Dengan adanya sistem informasi pengendalian persediaan bahan baku, proses pengecekan bahan baku dan laporan stok persediaan bahan baku dapat dibuat dengan waktu yang singkat.
2. Dengan adanya sistem informasi pengendalian persediaan bahan baku laporan pengolahan data *material list*, *purchase requisition*, dan *purchase order* menjadi terintegrasi dan mudah dipahami.

6.2. Saran

Saran yang dapat diberikan untuk pengembangan sistem informasi perencanaan produksi yang diusulkan ini selanjutnya adalah sebagai berikut:

1. Mengimplementasikan hasil dari penelitian berupa sistem informasi pengendalian persediaan bahan baku pada departemen *Inventory Material* PT Coca Cola Amatil Indonesia.
2. Diharapkan aplikasi ini dapat digunakan sebagai acuan dalam mengembangkan aplikasi-aplikasi lainnya agar sistem informasi pada PT Coca Cola Amatil Indonesia dapat lebih terintegrasi.

DAFTAR PUSTAKA

- Assauri, sofyan, 2004, *Manajemen produksi dan operasi* (Edisi revisi). Fakultas Ekonomi Universitas Indonesia. Jakarta.
- Assauri, Sofjan. 2010. *Manajemen Pemasaran*. Rajawali pers. Jakarta
- Harahap, Sofyan Syafri, 2013, *Analisis Kritis Atas Laporan Keuangan*, Cetakan Kesebelas, Penerbit Rajawali Pers, Jakarta.
- Jogiyanto H.M. 2005. *Analisis & Desain Sistem Informasi : Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis*. Yogyakarta. Andi.
- Kadir, Abdul. 2008. *Tuntunan Praktis Belajar Database Menggunakan MySQL*, C.V Andi Offset. Yogyakarta.
- Kristanto, Harianto. 2004. *Konsep & Perancangan Database*, Andi Offset, Yogyakarta.
- McLeod. 2004. *Sistem Informasi Manajemen*. PT. Indeks. Jakarta.
- McLeod, R. J, & Scholl. 2008. *Sistem Informasi Manajemen*. Edisi 10. Jakarta: Salemba Empat.
- Munawar. 2005. *Pemodelan dengan UML*. Yogyakarta: Graha Ilmu.
- Nugroho, Adi. (2007). *Rekayasa Perangkat Lunak Menggunakan UML dan Java*. ANDI. Yogyakarta.
- Riyanto, Agus. 2009. *Aplikasi Metodologi Penelitian Kesehatan*. Yogyakarta. Nuha Medika
- Robbin, J dan Coulter, W, 2002, *Fungsi Perencanaan dan Pengambilan Keputusan*. Jakarta
- Rosa dan Shalahuddin, M. 2011. *Rekayasa Perangkat Lunak*. Bandung: Modula.
- Sidik. 2004. *Pemrograman Web dengan PHP*, Informatika, Bandung.
- Sommerville, I. 2003. *Software Engineering RPL*. Edisi Keenam. Jakarta: Erlangga.
- Sutanta, Eddy. 2003. *Sistem Informasi Manajemen*. Graha Ilmu. Yogyakarta.

Tata Sutabri. 2003. *Analisa Sistem Informasi*. Penerbit Andi Yogyakarta.

Terry FitzGerald, Ardra F. FitzGerald, Warren D. Stallings, Jr. 2001. *Fundamentals of System Analysis*. John Wiley & Sons. New York.

Wilson, I.G. & Wilson, M.E. (1982). *Manajemen, Innovation, & System Design*. Canada: Auerbach Publishers Inc.

Zulfikarijah, Fien. *Manajemen Persediaan*. Malang. 2005.

Lampiran A



APLIKASI PENGENDALIAN PERSEDIAAN BAHAN BAKU PT. COCA COLA AMATIL INDONESIA



DATA MASTER

Data User

Data Supplier

Data Bahan Baku

TRANSAKSI

Material List

Purchase Requisition

Purchase Order

Barang Masuk

Barang Keluar

LAPORAN

Laporan Barang Masuk

Halaman Utama

Laporan Stok Bahan Baku

Barang Keluar

Material List

Data Material List

Data Bahan Baku

Data User

Data Supplier

DATA USER

Tambah Edit Hapus

ID USER	NAMA USER	LEVEL	HAK AKSES
1 123	Inventory mater	inv.material	supplier, barang, user, material list, purchase requisition, purchase order, barang masuk, barang k
2 34	Gudang	gudang	barang masuk, barang keluar, laporan barang masuk, laporan barang keluar, request material list
3 35	Admin	ppic	supplier, barang, user, purchase requisition, purchase order, laporan stock
4 789	Produksi	produksi	material list

10

Page 1 of 1

Displaying 1 to 4 of 4 items

APLIKASI PENGENDALIAN PERSEDIAAN BAHAN BAKU PT. COCA COLA AMATIL INDONESIA



DATA MASTER

Data User

Data Supplier

Data Bahan Baku

TRANSAKSI

Material List

Purchase Requisition

Purchase Order

Barang Masuk

Barang Keluar

LAPORAN

Laporan Barang Masuk

Halaman Utama

Laporan Stok Bahan Baku

Barang Keluar

Material List

Data Material List

Data Bahan Baku

Data User

Data Supplier

DATA SUPPLIER

Tambah Edit Hapus

SUPPLIER ID	SUPPLIER NAME	ADDRESS	PHONE NUMBER
1 SUP - 02	PT.AIRINDO PERKASA	jatinegara, cakung	622189107684
2 SUP - 03	PT.INDOTRADING	KARANG SATRIA	622189102265
3 SUP - 04	PT.CITRANATA PRAMANA	menteng, jaktim	622146834585
4 SUP - 05	PT.ROHA LAUTAN PEWARNA	kalirungkut,surabaya	623184801737

10

Page 1 of 1

Displaying 1 to 4 of 4 items

APLIKASI PENGENDALIAN PERSEDIAAN BAHAN BAKU PT. COCA COLA AMATIL INDONESIA



Material List

Purchase Requisition

Purchase Order

Barang Masuk

Barang Keluar

LAPORAN

Laporan Barang Masuk

Laporan Barang Keluar

Laporan Stok Bahan Baku

Request Material List

Logout

Halaman Utama

Material List

MATERIAL LIST

Ketik Nama Barang : Cari Atau tampilkan berdasarkan data sebelumnya : pilih kode material list --

MATERIAL ID	MATERIAL NAME	QTY	CATAT
			OK

DATA MATERIAL LIST

Hapus Data

MATERIAL ID	MATERIAL NAME	UNIT	QTY
-------------	---------------	------	-----

10

Page 1 of 1

Displaying 0 to 0 of 0 item

APLIKASI PENGENDALIAN PERSEDIAAN BAHAN BAKU
PT. COCA COLA AMATIL INDONESIA



Material List

Purchase Requisition

Purchase Order

Barang Masuk

Barang Keluar

LAPORAN

Laporan Barang Masuk

Laporan Barang Keluar

Laporan Stok Bahan Baku

Request Material List

Logout

Halaman UtamaPurchase Requisition x

PURCHASE REQUISITION

Pilih Kode Material List: -- pilih kode material list -- Tambah Atau tampilkan berdasarkan PR sebelumnya: -- pilih kode pr -- Tampilkan

DATA BARANG MASUK

Hapus Data

MATERIAL ID	MATERIAL NAME	UNIT	STOCK
-------------	---------------	------	-------

10 Page 1 of 1 Displaying 0 to 0 of 0 item

Id Transaksi: PR2016041300104 From: warehouse To: purchasing Simpan dan cetak Simpan

APLIKASI PENGENDALIAN PERSEDIAAN BAHAN BAKU
PT. COCA COLA AMATIL INDONESIA



Material List

Purchase Requisition

Purchase Order

Barang Masuk

Barang Keluar

LAPORAN

Laporan Barang Masuk

Laporan Barang Keluar

Laporan Stok Bahan Baku

Request Material List

Logout

Halaman UtamaPurchase Order x

PURCHASE ORDER

Ketik Kode PR: -- pilih kode pr -- Tambah Atau tampilkan berdasarkan PO sebelumnya: -- pilih kode po -- Tampilkan

Data Barang PURCHASE ORDER

Hapus Data

MATERIAL ID	MATERIAL NAME	UNIT	STOCK	QTY
-------------	---------------	------	-------	-----

10 Page 1 of 1 Displaying 0 to 0 of 0 item

Id Transaksi: PO2016041300010 From: warehouse To: purchasing Simpan dan cetak Simpan

Material List
Purchase Requisition
Purchase Order
Barang Masuk
Barang Keluar

Halaman Utama
Barang Masuk

CATAT BARANG MASUK

Pilih Kode PO:
-- pilih kode PO --
Pilih

Atau tampilkan berdasarkan transaksi sebelumnya:
-- pilih kode transaksi --
Tampilkan

DATA BARANG MASUK

Edit Data
Hapus Data

MATERIAL ID	MATERIAL NAME	QTY	UNIT	PRICE	SUB TOTAL

10

10

Page 1 of 1

10

10

Id Transaksi: IN2016041300065
No PO:
Id Supplier: PTAIRINDO PERKASA
Simpan dan cetak
Simpan

Displaying 0 to 0 of 0 items

Material List

Purchase Requisition

Purchase Order

Barang Masuk

Barang Keluar

Halaman Utama

Barang Keluar

CATAT BARANG KELUAR

Ketik Nama Barang : Cari Atau tampilkan berdasarkan transaksi sebelumnya: -- pilih kode transaksi -- Tampilkan

MATERIAL ID	MATERIAL NAME	PRICE	STOCK	QTY	CATAT
					OK

LAPORAN

Laporan Barang Masuk

Laporan Barang Keluar

Laporan Stok Bahan Baku

Request Material List

Logout

DATA BARANG KELUAR

Edit Data Hapus Data

MATERIAL ID	MATERIAL NAME	QTY	UNIT

10 Page 1 of 1

Displaying 0 to 0 of 0 items

Material List
Purchase Requisition
Purchase Order
Barang Masuk
Barang Keluar

LAPORAN

Laporan Barang Masuk
Laporan Barang Keluar
Laporan Stok Bahan Baku
Request Material List

Logout

Data Bahan Baku <
Data User <
Data Supplier <
Purchase Order <
Purchase Requisition <
Barang Masuk <
Laporan Barang Keluar <
Laporan Barang Masuk <

LAPORAN BARANG MASUK

Pilih periode tanggal laporan melalui datebox.

Tanggal Awal: Tanggal Akhir:

DATA BARANG MASUK							
ID TRANSAKSI	TANGGAL	SUPPLIER	MATERIAL ID	MATERIAL NAME	JML MASUK	PRICE	SUB TOTAL
PDF							

APLIKASI PENGENDALIAN PERSEDIAAN BAHAN BAKU PT. COCA COLA AMATIL INDONESIA



Material List
Purchase Requisition
Purchase Order
Barang Masuk
Barang Keluar

LAPORAN
Laporan Barang Masuk
Laporan Barang Keluar
Laporan Stok Bahan Baku
Request Material List

Logout

Data Bahan Baku x Data User x Data Supplier x Purchase Order x Purchase Requisition x Barang Masuk x Laporan Barang Keluar x Laporan Barang Masuk x

LAPORAN BARANG KELUAR

Pilih periode tanggal laporan melalui datebox.

Tanggal Awal: Tanggal Akhir:

ID TRANSAKASI	TANGGAL	MATERIAL ID	MATERIAL NAME	PRICE	JML KELUAR	SUB TOTAL
PDF						

APLIKASI PENGENDALIAN PERSEDIAAN BAHAN BAKU PT. COCA COLA AMATIL INDONESIA



Material List
Purchase Requisition
Purchase Order
Barang Masuk
Barang Keluar

LAPORAN
Laporan Barang Masuk
Laporan Barang Keluar
Laporan Stok Bahan Baku
Request Material List

Logout

Halaman Utama x Barang Keluar x Laporan Stok Bahan Baku x

LAPORAN STOCK MATERIAL LIST

MATERIAL ID	MATERIAL NAME	PRICE	STOCK	UNIT
BBW00001	Pewarna	9500	200	liter
BBW00002	Air	11000	1500	liter
BBW00003	Konsentrat	7000	2500	liter
BBW00004	Perisa	6500	2000	liter
BBW00005	Gula	9000	4000	kg
BBW00006	Bulir Jeruk	8000	3000	kg

APLIKASI PENGENDALIAN PERSEDIAAN BAHAN BAKU PT. COCA COLA AMATIL INDONESIA



Material List
Purchase Requisition
Purchase Order
Barang Masuk
Barang Keluar

LAPORAN
Laporan Barang Masuk
Laporan Barang Keluar
Laporan Stok Bahan Baku
Request Material List

Logout

Halaman Utama x Data Material List x

DAFTAR REQUEST MATERIAL LIST

Pilih Kode Material List:

ID MATERIAL LIST	TANGGAL	BANYAK BARANG	SUDAH DI CEK
ML2016041300026	2016-04-13	3	Y Lihat
ML2016041300025	2016-04-13	4	Y Lihat
ML2016041300024	2016-04-13	4	Y Lihat
ML2016041100023	2016-04-11	2	Y Lihat
ML2016041100022	2016-04-11	3	Y Lihat
ML2016041100021	2016-04-11	1	Y Lihat
ML2016041100020	2016-04-11	2	Y Lihat
ML2016041100019	2016-04-11	1	Y Lihat
ML2016041100018	2016-04-11	1	Y Lihat
ML2016040600017	2016-04-06	2	Y Lihat
ML2016040600016	2016-04-06	1	Y Lihat
ML2016032300015	2016-03-23	1	Y Lihat
ML2016032300014	2016-03-23	1	Y Lihat
ML2016031800013	2016-03-18	2	Y Lihat

Lampiran B

Koneksi

```
<?php
```

```
define('DB_HOST', 'localhost');  
define('DB_USERNAME', 'root');  
define('DB_PASSWORD', '');  
define('DB_NAME', 'eoq');
```

```
require(__DIR__.'/DB.php');
```

```
$koneksi = mysql_connect(DB_HOST, DB_USERNAME, DB_PASSWORD);  
$pilih_db = mysql_select_db(DB_NAME);
```

```
DB::initialize();
```

```
?>
```

Index

```
<?php
```

```
session_start();
```

```
if(empty($_SESSION['id_user'])&&empty($_SESSION['nm_level'])){
```

```
header("location:login.php");
```

```
}
```

```
else{
```

```
?>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
```

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
    <head>
```

```
        <meta http-equiv="Content-Type" content="text/html; charset=utf-
```

```
8" />
```

```

<title>APLIKASI PENGENDALIAN PERSEDIAAN BAHAN
BAKU PT COCA COLA AMATIL INDONESIA</title>

<link rel="stylesheet" type="text/css" href="mycss/index.css">
<link rel="stylesheet" type="text/css"
href="jquery_easyui/themes/default/easyui.css">
<link rel="stylesheet" type="text/css"
href="jquery_easyui/themes/icon.css">
<link rel="stylesheet" type="text/css"
href="jquery_easyui/themes/panel.css" />
<link rel="shortcut icon" href="mycss/logo.png" />
<script type="text/javascript"
src="jquery_easyui/jquery.min.js"></script>
<script type="text/javascript"
src="jquery_easyui/jquery.easyui.min.js"></script>
<script>
function addTab(title, url){
    if ($('#tt').tabs('exists', title)){
        $('#tt').tabs('select', title);
    } else {
        var content = '<iframe scrolling="auto"
frameborder="0" src="'+url+'" style="width:100%;height:100%;"></iframe>';
        $('#tt').tabs('add',{
            title:title,
            content:content,
            closable:true
        });
    }
}
</script>
</head>
<body>

```

```

<div id="header">
</div>
<div id="navigasi">
    <div style="width:200px;height:auto;padding:5px;float:left;
margin-right:10px;background:#e6e6e6;">
        <div class="easyui-panel" title="DATA MASTER"
collapsible="true" style="width:200px;height:auto;padding:10px;">
            <?php if('admin' == $_SESSION['level'] OR
in_array('user',$_SESSION['hak_akses'])):?>
                <a href="#" class="easyui-linkbutton"
plain="true" onClick="addTab('Data User','user.php')">Data User</a><br />
            <?php endif;?>
            <?php if('admin' == $_SESSION['level'] OR
in_array('supplier',$_SESSION['hak_akses'])):?>
                <a href="#" class="easyui-linkbutton"
plain="true" onClick="addTab('Data Supplier','supplier.php')">Data
Supplier</a><br />
            <?php endif;?>
            <?php if('admin' == $_SESSION['level'] OR
in_array('barang',$_SESSION['hak_akses'])):?>
                <a href="#" class="easyui-linkbutton"
plain="true" onClick="addTab('Data Bahan Baku','barang.php')">Data Bahan
Baku</a><br />
            <?php endif;?>
        </div>
    <br/>
    <div class="easyui-panel" title="TRANSAKSI"
collapsible="true" style="width:200px;height:auto;padding:10px;">
        <?php if(in_array($_SESSION['level'],
array('admin','inv.material','inv.manager'))):?>

```

```

<a href="#" class="easyui-linkbutton"
plain="true" onClick="addTab('Purchase
Requisition','purchase_requisition.php')">Purchase Requisition</a>
<?php endif;?>
<?php if(in_array($_SESSION['level'],
array('admin','inv.material','inv.manager'))):?>
<a href="#" class="easyui-linkbutton"
plain="true" onClick="addTab('Purchase
Order','laporan/stok_barang/pilih_lap.php')">Purchase Order</a>
<?php endif;?>
<?php if('admin' == $_SESSION['level'] OR
in_array('barang masuk',$_SESSION['hak_akses'])):?>
<a href="#" class="easyui-linkbutton"
plain="true" onClick="addTab('Barang Masuk','barang_masuk.php')">Barang
Masuk</a><br />
<?php endif;?>
<?php if('admin' == $_SESSION['level'] OR
in_array('barang keluar',$_SESSION['hak_akses'])):?>
<a href="#" class="easyui-linkbutton"
plain="true" onClick="addTab('Barang Keluar','barang_keluar.php')">Barang
Keluar</a><br />
<?php endif;?>
<?php if('admin' == $_SESSION['level'] OR
in_array('eoq',$_SESSION['hak_akses'])):?>
<a href="#" class="easyui-linkbutton"
plain="true" onClick="addTab('Hitung EOQ','eoq.php')">Hitung EOQ</a><br />
<?php endif;?>

</div><br />

```

```

<div class="easyui-panel" title="LAPORAN"
collapsible="true" style="width:200px;height:auto;padding:10px;">
    <?php if('admin' == $_SESSION['level'] OR
in_array('laporan barang masuk',$_SESSION['hak_akses'])):?>
        <a href="#" class="easyui-linkbutton"
plain="true" onClick="addTab('Laporan Barang
Masuk','laporan/barang_masuk/pilih_lap.php')">Laporan Barang Masuk</a><br
/>
        <?php endif;?>
        <?php if('admin' == $_SESSION['level'] OR
in_array('laporan barang keluar',$_SESSION['hak_akses'])):?>
            <a href="#" class="easyui-linkbutton"
plain="true" onClick="addTab('Laporan Barang
Keluar','laporan/barang_keluar/pilih_lap.php')">Laporan Barang Keluar</a><br
/>
            <?php endif;?>
            <?php if(in_array($_SESSION['level'],
array('admin','inv.material'))):?>
                <a href="#" class="easyui-linkbutton"
plain="true" onClick="addTab('Laporan
EOQ','laporan/eq/pilih_lap.php')">Laporan EOQ</a><br />
                <?php endif;?>
                <?php if(in_array($_SESSION['level'],
array('admin','inv.material','produksi'))):?>
                    <a href="#" class="easyui-linkbutton"
plain="true" onClick="addTab('Laporan Stok Bahan
Baku','laporan/stok_barang/lihat.php')">Laporan Stok Bahan Baku</a><br />
                    <?php endif;?>
            </div>
            <div class="easyui-panel" title=""
collapsible="true" style="width:200px;height:auto;padding:10px;">

```

```

        <a href="akses/logout.php" class="easyui-
linkbutton" iconCls="icon-logout" plain="true">Logout</a>
    </div>
</div>
</div>
<div id="isi">
    <div id="tt" class="easyui-tabs" style="height:500px;">
        <div title="Halaman Utama" style="padding-
top:20px; text-align:center; background-image:url(mycss/images/materials.png);
background-repeat:no-repeat;background-color:#FFF;">
            </div>
        </div>
    </div>
</div>
<div id="footer">
    Aplikasi Pengendalian Persediaan Bahan Baku Dengan
    Metode EOQ| copyright Muhammad Alam Muntaha | 2015
</div>
</body>
</html>
<?php } ?>

```

Barang

```

<?php
    define('CORE_FILE', __FILE__);
    require("config.php");

    include("koneksi/koneksi.php");
    $q_list_supplier = mysql_query("SELECT * FROM supplier");
    $list_supplier = array();
    while($sup = mysql_fetch_assoc($q_list_supplier)) {
        $list_supplier[] = $sup;
    }
}

```

```

    }
?>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=utf-
8" />

        <title>Untitled Document</title>
        <link rel="stylesheet" type="text/css"
href="jquery_easyui/themes/default/easyui.css">
        <link rel="stylesheet" type="text/css"
href="jquery_easyui/themes/icon.css">
        <link rel="stylesheet" type="text/css"
href="jquery_easyui/themes/style.css">
        <script type="text/javascript"
src="jquery_easyui/jquery.min.js"></script>
        <script type="text/javascript"
src="jquery_easyui/jquery.easyui.min.js"></script>
        <script type="text/javascript" src="libs_js/barang.js"></script>
    </head>
    <body>
        <h2>DATA BAHAN BAKU</h2>
        <table id="dg" title="DATA BAHAN BAKU" class="easyui-
datagrid" style="height:250px"
            url="data_master/barang/get_barang.php"
            toolbar="#toolbar" pagination="true"
            rownumbers="true" fitColumns="true"
singleSelect="true">
            <thead>
                <tr>

```

```

        <th field="id_barang" width="30">ID
BARANG</th>
        <th field="nm_barang" width="40">NAMA
BARANG</th>
        <th field="nm_supplier"
width="40">SUPPLIER</th>
        <th field="stok" width="20">STOK</th>
        <th field="stok_minimum"
width="20">STOK MIN</th>
        <th field="satuan"
width="30">SATUAN</th>
        <th field="hrg_brg" width="30">HARGA
BARANG</th>
        <th field="hrg_smpn" width="30">BIAYA
SIMPAN</th>
        <th field="hrg_pesn" width="30">BIAYA
PESAN</th>
        <th field="permintaan"
width="40">PERMINTAAN PERTAHUN</th>
    </tr>
</thead>
</table>
<div id="toolbar">
    <a href="#" class="easyui-linkbutton" iconCls="icon-add"
plain="true" onclick="newData()">Tambah</a>
    <a href="#" class="easyui-linkbutton" iconCls="icon-edit"
plain="true" onclick="editData()">Edit</a>
    <a href="#" class="easyui-linkbutton" iconCls="icon-
remove" plain="true" onclick="removeData()">Hapus</a>
</div>

```

```

<div id="dlg" class="easyui-dialog"
style="width:400px;height:340px;padding:10px 20px"
closed="true" buttons="#dlg-buttons">
<div class="ftitle">Informasi Barang</div>
<form id="fm" method="post" novalidate>
<div class="fitem">
<label>Id Barang:</label>
<input name="id_barang" id="id_barang"
class="easyui-validatebox" required="true">
</div>
<div class="fitem">
<label>Nama Barang:</label>
<input name="nm_barang" id="nm_barang"
class="easyui-validatebox" required="true">
</div>
<div class="fitem">
<label>Stok:</label>
<input name="stok" class="easyui-
numberbox" required="true" >
</div>
<div class="fitem">
<label>Stok Minimum:</label>
<input name="stok_minimum"
class="easyui-numberbox" required="true" >
</div>
<div class="fitem">
<label>Satuan:</label>
<!--
<input name="satuan" id="satuan"
class="easyui-validatebox" required="true">
-->

```

```

        <select class="easyui-selectbox"
name="satuan" style="width:150px;" required="true">
        <?php foreach($configs['satuan'] as
$satunya => $max):?>
        <option value="<?php echo
$satunya;?>"><?php echo $satunya;?></option>
        <?php endforeach;?>
        </select>
    </div>
    <div class="fitem">
        <label>Harga barang:</label>
        <input type="number" id="harga-barang"
name="hrg_brg" class="easyui-numberbox" required="true" min="0" value="0">
    </div>
    <div class="fitem">
        <label>Biaya simpan:</label>
        <input type="number" id="biaya-simpan"
name="hrg_smpn" class="easyui-numberbox" required="true" min="0"
value="0">
    </div>
    <div class="fitem">
        <label>Biaya pesan:</label>
        <input type="number" id="biaya-pesan"
name="hrg_pesan" class="easyui-numberbox" required="true" min="0"
value="0">
    </div>
    <div class="fitem">
        <label>Permintaan:</label>
        <input name="permintaan" class="easyui-
numberbox" required="true" >
    </div>

```

```

        <div class="fitem">
            <label>Supplier:</label>
            <select name="id_supplier" class="easyui-
selectbox" required="true" >
                <?php foreach($list_supplier as $sup):?>
                    <option value="<?php echo
$sup['id_supplier'];?>"><?php echo $sup['nm_supplier'];?></option>
                <?php endforeach;?>
            </select>
        </div>
    </form>
</div>
<div id="dlg-buttons">
    <a href="#" class="easyui-linkbutton" iconCls="icon-ok"
onclick="saveData()">Save</a>
    <a href="#" class="easyui-linkbutton" iconCls="icon-
cancel" onclick="javascript:$('#dlg').dialog('close')">Cancel</a>
</div>
</body>
</html>

```

Login Error

```

<?php
// Warning Error To Login Admin Page
$error_login = "Maaf, Username & Password Salah!.";
// View Error Message To Browser
echo "
<html>
<head>
<title>Login Administrator</title>
<link rel="stylesheet" type="text/css" href="../css/style_login.css" />
<link rel="shortcut icon" href="../images/images_admin/favicon.ico" />

```

```

</head><body><div id=\"main\" style=\"width:560px;\">
<div id=\"error_login\"><img src=\"../images/images_login/img_login_lock.png\"
width=\"30\" height=\"31\" align=\"absmiddle\" class=\"img_lock\"/>
$error_login
<br /><center><a href=\"login.php\" class=\"clickhere\">ULANGI
LAGI</a></center>
</div> <div class=\"clear\"></div>
<div id=\"vertical_effect\">&nbsp;</div>
</div></body></html>";?>

```

Logout

```

<?php
session_start();
unset($_SESSION['id_user']); unset($_SESSION['akses']);
session_destroy(); header('location:login.php');
?>

```

Barang Keluar

```

<?php

include("koneksi/koneksi.php");

$query_list_nama_barang = mysql_query("select nm_barang from
barang") or die(mysql_error());

$list_nama_barang = array();

while($brg = mysql_fetch_assoc($query_list_nama_barang)) {

    $list_nama_barang[] = $brg['nm_barang'];

}

$query_suppliers = mysql_query("select * from supplier");

$suppliers = [];

while($supplier = mysql_fetch_assoc($query_suppliers)) {

```

```

        $suppliers[] = $supplier;
    }

?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

    <head>

        <meta http-equiv="Content-Type" content="text/html; charset=utf-
8" />

        <title>Untitled Document</title>

        <link rel="stylesheet" type="text/css"
href="jquery_easyui/themes/default/easyui.css">

        <link rel="stylesheet" type="text/css"
href="jquery_easyui/themes/icon.css">

        <link rel="stylesheet" type="text/css"
href="jquery_easyui/themes/style.css">

        <link rel="stylesheet" type="text/css" href="mycss/request.css" />

        <script type="text/javascript"
src="jquery_easyui/jquery.min.js"></script>

        <script type="text/javascript"
src="jquery_easyui/jquery.easyui.min.js"></script>

        <script type="text/javascript"
src="jquery_easyui/jquery.form.js"></script>

        <script type="text/javascript"
src="libs_js/barang_keluar.js"></script>

```

```

</head>

<body>

    <div id="cari">

        <form action="transaksi/search_barang_by_nama.php"
method="post" id="form_cari">

            Ketik Nama Barang :

            <input name="cari_id_brg" type="text"
id="cari_id_brg" list="list_nama_barang"/>

            <datalist id="list_nama_barang">

                <?php foreach($list_nama_barang as
$id_brg):?>

                    <option value="<?php echo $id_brg;?>"/>

                <?php endforeach;?>

            </datalist>

            <input type="submit" name="cari" id="cari2"
value="Cari" />

        </form>

    </div>

    <div id="title">CATAT BARANG KELUAR</div>

    <div id="data">

        <form action="transaksi/lap_barang_keluar/save_item.php"
method="post" name="form_item" id="form_item">

            <table width="400" border="1" align="center"
cellpadding="0" cellspacing="0" id="tabel_item">

                <tr class="tr_item">

```

	ID
BARANG	
	NAMA
BARANG	
	HARGA
BARANG	
	STOK
	JUMLAH
KELUAR	
	CATAT

<input disabled="disabled" id="id_barang" readonly="readonly" size="20" type="text"/>
<input disabled="disabled" id="nm_barang" readonly="readonly" size="20" type="text"/>
<input disabled="disabled" id="hrg_brg" readonly="readonly" size="20" type="text"/>
<input disabled="disabled" id="stok" readonly="readonly" size="10" type="text"/>
<input disabled="disabled" id="jumlah" size="10" type="text"/>
<input id="tambah_item" name="tambah_item" type="submit" value="OK"/>

```

        </tr>

    </table>

</form>

</div>

<div id="detail_request">

    <table id="dg" title="DATA BARANG KELUAR"
class="easyui-datagrid" style="height:250px"

        url="transaksi/lap_barang_keluar/get_item.php"

        toolbar="#toolbar" pagination="true"

        rownumbers="true" fitColumns="true"

singleSelect="true">

        <thead>

            <tr>

                <th field="id_barang"

width="30">ID BARANG</th>

                <th field="nm_barang"

width="30">NAMA BARANG</th>

                <th field="jml"

width="30">JUMLAH</th>

                <th field="satuan"

width="30">SATUAN</th>

                <!--

                <th field="hrg_brg"

width="30">HARGA</th>

                <th field="sub_total"

width="50">SUB TOTAL</th>

```

```

-->

</tr>

</thead>

</table>

<div id="toolbar">

    <a href="#" class="easyui-linkbutton"
iconCls="icon-edit" plain="true" onclick="editData()">Edit Data</a>

    <a href="#" class="easyui-linkbutton"
iconCls="icon-remove" plain="true" onclick="removeData()">Hapus Data</a>

</div>

<div id="dlg" class="easyui-dialog"
style="width:400px;height:280px;padding:10px 20px"

closed="true" buttons="#dlg-buttons">

    <div class="fitem">

        <label>Id Barang:</label>

        <input name="id_barang"
disabled="disabled" class="easyui-validatebox" >

    </div>

    <div class="fitem">

        <label>Nama Barang:</label>

        <input name="nm_barang"
disabled="disabled" class="easyui-validatebox" >

    </div>

    <div class="fitem">

        <label>Harga:</label>

```

```
                <input name="hrg_brg" class="easyui-  
numberbox" readonly="readonly" required="true" >
```

```
            </div>
```

```
            <div class="fitem">
```

```
                <label>Jml Keluar:</label>
```

```
                <input name="jml" class="easyui-  
numberbox" required="true" >
```

```
            </div>
```

```
            <div class="fitem">
```

```
                <label>Satuan:</label>
```

```
                <input name="satuan" class="easyui-  
numberbox" required="true" >
```

```
            </div>
```

```
        </form>
```

```
    </div>
```

```
    <div id="dlg-buttons">
```

```
        <a href="#" class="easyui-linkbutton" iconCls="icon-ok"  
onclick="saveData()">Save</a>
```

```
        <a href="#" class="easyui-linkbutton" iconCls="icon-  
cancel" onclick="javascript:$('#dlg').dialog('close')">Cancel</a>
```

```
    </div>
```

```
<div id="proses">
```

```
<!--
```

```

        <input type='checkbox' name='term'
onClick="Javascript:disab(this, 1);"/>Selesai Menambahkan<br /><br />
        -->
        <form action="transaksi/lap_barang_keluar/proses.php"
method="post" id="form_proses">
            <label>Id Transaksi:</label>
            <input name="id_trx" type="text" id="id_trx"
readonly="readonly" />
            <input type="submit" name="proses_request"
id="proses_request" value="simpan dan cetak" />
            <input type="submit" name="proses_request"
id="proses_request" value="simpan" />
        </form>
    </div>
</div>
</body>
</html>

```

Barang Masuk

```

<?php

include("koneksi/koneksi.php");

```

```
$query_list_id_eoq = mysql_query("select id_transaksi from eoq") or  
die(mysql_error());
```

```
$list_id_eoq = array();
```

```
while($brg = mysql_fetch_assoc($query_list_id_eoq)) {
```

```
    $list_id_eoq[] = $brg['id_transaksi'];
```

```
}
```

```
$query_suppliers = mysql_query("select * from supplier");
```

```
$suppliers = [];
```

```
while($supplier = mysql_fetch_assoc($query_suppliers)) {
```

```
    $suppliers[] = $supplier;
```

```
}
```

```
?>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
```

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
    <head>
```

```
        <meta http-equiv="Content-Type" content="text/html; charset=utf-  
8" />
```

```
        <title>Untitled Document</title>
```

```
        <link rel="stylesheet" type="text/css"  
href="jquery_easyui/themes/default/easyui.css">
```

```
        <link rel="stylesheet" type="text/css"  
href="jquery_easyui/themes/icon.css">
```

```

        <link rel="stylesheet" type="text/css"
href="jquery_easyui/themes/style.css">

        <link href="mycss/request.css" rel="stylesheet" type="text/css" />

        <script type="text/javascript"
src="jquery_easyui/jquery.min.js"></script>

        <script type="text/javascript"
src="jquery_easyui/jquery.easyui.min.js"></script>

        <script type="text/javascript"
src="jquery_easyui/jquery.form.js"></script>

        <script type="text/javascript"
src="libs_js/barang_masuk.js"></script>

    </head>

    <body>

        <div id="cari">

            <form
action="transaksi/barang_masuk/get_eoq_items.php" method="post"
id="form_cari">

```

Ketik No EOQ :

```

        <input name="id_eoq" type="text" id="id_eoq"
list="list_id_eoq"/>

        <datalist id="list_id_eoq">

            <?php foreach($list_id_eoq as
$nm_barang):?>

                <option value="<?php echo
$nm_barang;?>" />

            <?php endforeach;?>

```

```

        </datalist>

        <input type="submit" name="cari" id="cari2"
value="Cari" />

    </form>

</div>

<!-- <div id="title">CATAT BARANG MASUK</div>

<div id="data">

    <table width="400" border="1" align="center"
cellpadding="0" cellspacing="0" id="tabel_item">

        <tr class="tr_item">

            <td align="center">ID BARANG</td>

            <td align="center">NAMA BARANG</td>

            <td align="center">HARGA
BARANG</td>

            <td align="center"> STOK</td>

            <td align="center">JUMLAH MASUK</td>

            <td align="center">CATAT</td>

        </tr>

        <form
action="transaksi/lap_barang_masuk/save_item.php" method="post"
name="form_item" id="form_item">

            <tr>

```

```

<td><input name="id_barang"
type="text" disabled="disabled" id="id_barang" size="20" readonly="readonly"
/></td>

<td><input name="nm_barang"
type="text" disabled="disabled" id="nm_barang" size="20" readonly="readonly"
/></td>

<td><input name="hrg_brg"
type="text" disabled="disabled" id="hrg_brg" size="20" readonly="readonly"
/></td>

<td><input name="stok" type="text"
disabled="disabled" id="stok" size="10" readonly="readonly" /></td>

<td><input name="jumlah"
type="text" disabled="disabled" id="jumlah" size="10" /></td>

<td><input type="submit"
name="tambah_item" id="tambah_item" value="OK " /></td>

</tr>

</form>

```

```

</table>

</div> -->

<div id="detail_request">

<table id="dg" title="DATA BARANG MASUK"
class="easyui-datagrid" style="height:250px"

url="transaksi/lap_barang_masuk/get_temp_barang.php"

toolbar="#toolbar" pagination="true"

```

```

rownumbers="true" fitColumns="true"
singleSelect="true">

<thead>

<tr>

<th field="id_barang"
width="30">ID BARANG</th>

<th field="nm_barang"
width="30">NAMA BARANG</th>

<th field="jml_masuk"
width="30">JUMLAH</th>

<th field="satuan"
width="30">SATUAN</th>

<th field="hrg_beli"
width="30">HARGA</th>

<th field="sub_total"
width="50">SUB TOTAL</th>

</tr>

</thead>

</table>

<div id="toolbar">

<a href="#" class="easyui-linkbutton"
iconCls="icon-edit" plain="true" onclick="editData()">Edit Data</a>

<a href="#" class="easyui-linkbutton"
iconCls="icon-remove" plain="true" onclick="removeData()">Hapus Data</a>

</div>

```

```

<div id="dlg" class="easyui-dialog"
style="width:400px;height:280px;padding:10px 20px"

        closed="true" buttons="#dlg-buttons">

        <div class="ftitle">Edit Jumlah Masuk</div>

        <form id="fm" method="post" novalidate>

                <div class="fitem">

                        <label>Id Barang:</label>

                        <input name="id_barang"
disabled="disabled" class="easyui-validatebox" >

                </div>

                <div class="fitem">

                        <label>Nama Barang:</label>

                        <input name="nm_barang"
disabled="disabled" class="easyui-validatebox" >

                </div>

                <div class="fitem">

                        <label>Harga:</label>

                        <input name="hrg" class="easyui-
numberbox" readonly="readonly" required="true" >

                </div>

                <div class="fitem">

                        <label>Jml Masuk:</label>

                        <input name="jml" class="easyui-
numberbox" required="true" >

                </div>

```

```

        <div class="fitem">

            <label>Satuan:</label>

            <input name="satuan" class="easyui-
numberbox" required="true" >

        </div>

    </form>

</div>

<div id="dlg-buttons">

    <a href="#" class="easyui-linkbutton"
iconCls="icon-ok" onclick="saveData()">Save</a>

    <a href="#" class="easyui-linkbutton"
iconCls="icon-cancel" onclick="javascript:$('#dlg').dialog('close')>Cancel</a>

</div>

<div id="proses">

    <!--

    <input type='checkbox' name='term'
onClick="Javascript:disab(this, 1);"/>Selesai Menambahkan<br /><br />

    -->

    <template id="form-item-masuk">

        <input name="id_barang[]" type="hidden"
class="id_barang" size="15" readonly="readonly"/>

        <input name="stok_awal[]" type="hidden"
class="stok_awal" size="15" readonly="readonly"/>

        <input name="jml_masuk[]" type="hidden"
class="jml_masuk" size="15" readonly="readonly"/>

```

```

        <input name="hrg_beli[]" type="hidden"
class="hrg_beli" size="10" readonly="readonly"/>

        <input name="sub_total[]" type="hidden"
class="sub_total" size="15" readonly="readonly"/>

    </template>

    <form
action="transaksi/lap_barang_masuk/proses.php" method="post"
id="form_proses" style="">

        <label>Id Transaksi:</label>

        <input name="id_trx" type="text"
id="id_trx" readonly="readonly" />

        <label>No EOQ:</label>

        <input type="text" name="id_eoq"
id="no_eoq" readonly/>

        <label>Id Supplier:</label>

        <select name="id_supplier"
id="id_supplier" >

            <?php foreach($suppliers as
$supplier):?>

                <option value=""<?php echo
$supplier['id_supplier'];?>"><?php echo $supplier['nm_supplier'];?></option>

                <?php endforeach;?>

            </select>

            <input type="submit"
name="proses_request" id="proses_request" value="Simpan dan cetak"/>

            <input type="submit"
name="proses_request" id="proses_request" value="Simpan"/>

```

```
                <div id="form-items"></div>

            </form>

        </div>

    </div>

</body>

</html>
```

Config

```
<?php defined('CORE_FILE') or die('Access denied');
```

```
$configs = array(

    'hak_akses' => array(

        'supplier',

        'barang',

        'user',

        'purchase requisition',

        'purchase order',

        'barang masuk',

        'barang keluar',

        'laporan barang masuk',

        'laporan barang keluar',

        'laporan eq',

        'stok',

        'eq'
```

```

    ),
    'satuan' => array(
        'kg' => 100,
        'dus' => 20,
        'liter' => 50,
        'ml' => 10,
        'mg' => 10,
        /* 'kubik' => 50, */
    )
);

```

Login

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>Login Pengendalian Persediaan Bahan Baku Dengan Metode EOQ</title>

</a><link rel="stylesheet" type="text/css" href="mycss/login.css" />

<link rel="shortcut icon" href="mycss/logo.png" />

<script type="text/javascript" src="jquery_easyui/jquery.min.js"></script>

<script type="text/javascript" src="libs_js/login.js"></script>

</head>

```

```
<body>

<div class="lg-container">

    <h1>LOGIN USER</h1>

    <form id="lg-form" name="lg-form"
action="akses/p_login_gudang.php" method="post" onload="cekyangkosong();
">

        <div>

            <label for="username">ID User:</label>

            <input type="text" name="username" id="username"
placeholder="ID User"/>

        </div>

        <div>

            <label for="password">Password:</label>

            <input type="password" name="password"
id="password" placeholder="Password" />

        </div>

        <div>

            <button type="submit" id="login">Login</button>

        </div>

    </form>
```

```
<div id="message"></div>

</div>

</body>

</html>
```

Purchase Requisition

```
<?php

    include("koneksi/koneksi.php");

    $query_list_nama_barang = mysql_query("select nm_barang from
barang") or die(mysql_error());

    $list_nama_barang = array();

    while($brg = mysql_fetch_assoc($query_list_nama_barang)) {

        $list_nama_barang[] = $brg['nm_barang'];

    }

?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

    <head>

        <meta http-equiv="Content-Type" content="text/html; charset=utf-
8" />

        <title>Untitled Document</title>

        <link rel="stylesheet" type="text/css"
href="jquery_easyui/themes/default/easyui.css">

        <link rel="stylesheet" type="text/css"
href="jquery_easyui/themes/icon.css">
```

```

        <link rel="stylesheet" type="text/css"
href="jquery_easyui/themes/style.css">

        <link href="mycss/request.css" rel="stylesheet" type="text/css" />

        <script type="text/javascript"
src="jquery_easyui/jquery.min.js"></script>

        <script type="text/javascript"
src="jquery_easyui/jquery.easyui.min.js"></script>

        <script type="text/javascript"
src="jquery_easyui/jquery.form.js"></script>

        <script type="text/javascript" src="libs_js/pr.js"></script>

    </head>

    <body>

        <div id="cari">

            <form action="transaksi/search_barang_by_nama.php"
method="post" id="form_cari">

                Ketik Nama Barang :

                <input name="cari_id_brg" type="text"
id="cari_id_brg" list="list_nama_barang"/>

                <datalist id="list_nama_barang">

                    <?php foreach($list_nama_barang as
$nm_barang):?>

                        <option value="<?php echo
$nm_barang;?>" />

                    <?php endforeach;?>

                </datalist>

```

```
        <input type="submit" name="cari" id="cari2"
value="Cari" />
```

```
    </form>
```

```
</div>
```

```
<div id="title">PURCHASE REQUISITION</div>
```

```
<div id="data">
```

```
        <table width="400" border="1" align="center"
cellpadding="0" cellspacing="0" id="tabel_item">
```

```
        <tr class="tr_item">
```

```
            <td align="center">ID BARANG</td>
```

```
            <td align="center">NAMA BARANG</td>
```

```
            <td align="center">STOK</td>
```

```
            <td align="center">CATAT</td>
```

```
        </tr>
```

```
        <form action="transaksi/pr/save_item.php"
method="post" name="form_item" id="form_item">
```

```
        <tr>
```

```
            <td><input name="id_barang"
type="text" disabled="disabled" id="id_barang" size="20" readonly="readonly"
/></td>
```

```
            <td><input name="nm_barang"
type="text" disabled="disabled" id="nm_barang" size="20" readonly="readonly"
/></td>
```

```
            <td><input name="stok" type="text"
disabled="disabled" id="stok" size="10" readonly="readonly" /></td>
```

```
 <input type="submit" name="tambah_item" id="tambah_item" value="OK " /></td>  </tr>  </form>  </table>  </div>  <div id="detail_request">  <table id="dg" title="DATA BARANG MASUK" class="easyui-datagrid" style="height:250px"  url="transaksi/pr/get_temp_barang.php" toolbar="#toolbar" pagination="true" rownumbers="true" fitColumns="true" singleSelect="true">  <thead>  <tr>  <th field="id_barang" width="30">ID BARANG</th>  <th field="nm_barang" width="30">NAMA BARANG</th>  <th field="satuan" width="30">SATUAN</th>  <th field="stok" width="30">STOK</th>  </tr>  </thead>  </table> |
```

```

<div id="toolbar">

    <a href="#" class="easyui-linkbutton"
iconCls="icon-remove" plain="true" onclick="removeData()">Hapus Data</a>

</div>

<div id="proses">

    <!--

    <input type='checkbox' name='term'
onClick="Javascript:disab(this, 1);"/>Selesai Menambahkan<br /><br />

    -->

    <template id="form-pr-item">

        <input name="id_barang[]" type="hidden"
class="id_barang" size="15" readonly="readonly"/>

        <input name="jumlah[]" type="hidden"
class="jumlah" size="10" readonly="readonly"/>

    </template>

    <hr />

    <form action="transaksi/pr/proses.php"
method="post" id="form_proses" style="">

        <label>Id Transaksi:</label>

        <input name="id_trx" type="text"
id="id_trx" readonly="readonly" />

        <label>From:</label>

        <input name="from" type="text" id="from"
value="warehouse" readonly/>

        <label>To:</label>

```

```

                                <input name="to" type="text" id="to"
value="purchasing" readonly/>

                                <input type="submit"
name="proses_request" id="proses_request" value="Simpan dan cetak"/>

                                <input type="submit"
name="proses_request" id="proses_request" value="Simpan"/>

                                <div id="form-prs"></div>

                                </form>

                                </div>

                                </div>

                                </body>

</html>

```

Supplier

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>Untitled Document</title>

<link rel="stylesheet" type="text/css"
href="jquery_easyui/themes/default/easyui.css">

<link rel="stylesheet" type="text/css" href="jquery_easyui/themes/icon.css">

<link rel="stylesheet" type="text/css" href="jquery_easyui/themes/style.css">

```

```
<script type="text/javascript" src="jquery_easyui/jquery.min.js"></script>
<script type="text/javascript" src="jquery_easyui/jquery.easyui.min.js"></script>
<script type="text/javascript" src="libs_js/supplier.js"></script>
</head>
```

```
<body>
```

```
<h2>DATA SUPPLIER</h2>
```

```

    <table id="dg" title="DATA SUPPLIER" class="easyui-datagrid"
style="height:250px"

        url="data_master/supplier/get_supplier.php"

        toolbar="#toolbar" pagination="true"

        rownumbers="true" fitColumns="true"

singleSelect="true">

    <thead>

        <tr>

            <th field="id_supplier" width="50">ID SUPPLIER</th>

                <th field="nm_supplier" width="50">NAMA
SUPPLIER</th>

                <th field="almt_supplier"
width="50">ALAMAT</th>

                <th field="tlp_supplier"
width="50">TELEPON</th>

            </tr>

        </thead>
```

```
</table>
```

```
<div id="toolbar">
```

```
    <a href="#" class="easyui-linkbutton" iconCls="icon-add"
plain="true" onclick="newData()">Tambah</a>
```

```
    <a href="#" class="easyui-linkbutton" iconCls="icon-edit"
plain="true" onclick="editData()">Edit</a>
```

```
    <a href="#" class="easyui-linkbutton" iconCls="icon-remove"
plain="true" onclick="removeData()">Hapus</a>
```

```
</div>
```

```
<div id="dlg" class="easyui-dialog"
style="width:400px;height:310px;padding:10px 20px"
```

```
    closed="true" buttons="#dlg-buttons">
```

```
    <div class="ftitle">Informasi Supplier</div>
```

```
    <form id="fm" method="post" novalidate>
```

```
        <div class="fitem">
```

```
            <label>Id Supplier:</label>
```

```
            <input name="id_supplier" id="id_supplier"
class="easyui-validatebox" required="true">
```

```
        </div>
```

```
        <div class="fitem">
```

```
            <label>Nama Supplier:</label>
```

```
            <input name="nm_supplier" id="nm_supplier"
class="easyui-validatebox" required="true">
```

```
        </div>
```

```

<div class="fitem">

    <label>Alamat:</label>

    <textarea name="almt_supplier" cols="15" rows="3" class="easyui-
validatebox" id="almt_supplier" required="true"></textarea>

</div>

<div class="fitem">

    <label>Telepon:</label>

    <input name="tlp_supplier" class="easyui-numberbox"
id="stok" required="true" >

</div>

</form>

</div>

<div id="dlg-buttons">

    <a href="#" class="easyui-linkbutton" iconCls="icon-ok"
onclick="saveData()">Save</a>

    <a href="#" class="easyui-linkbutton" iconCls="icon-cancel"
onclick="javascript:$('#dlg').dialog('close')">Cancel</a>

</div>

</body>

</html>

```

User

```
<?php
```

```
define('CORE_FILE', __FILE__);

include "config.php";

?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

    <head>

        <meta http-equiv="Content-Type" content="text/html; charset=utf-
8" />

        <title>Untitled Document</title>

        <link rel="stylesheet" type="text/css"
href="jquery_easyui/themes/default/easyui.css">

        <link rel="stylesheet" type="text/css"
href="jquery_easyui/themes/icon.css">

        <link rel="stylesheet" type="text/css"
href="jquery_easyui/themes/style.css">

        <script type="text/javascript"
src="jquery_easyui/jquery.min.js"></script>

        <script type="text/javascript"
src="jquery_easyui/jquery.easyui.min.js"></script>

        <script type="text/javascript" src="libs_js/user.js"></script>

        <style type="text/css">

            .dropme {

                display: inline-block;

                clear: right;

                float: right;
```

```

        }

</style>

</head>

<body>

    <h2>DATA USER</h2>

    <table id="dg" title="DATA USER" class="easyui-datagrid"
style="height:250px"

        url="data_master/user/get_user.php"

        toolbar="#toolbar" pagination="true"

        rownumbers="true" fitColumns="true"

singleSelect="true">

        <thead>

            <tr>

                <th field="id_user" width="20">ID

USER</th>

                <th field="nm_user" width="20">NAMA

USER</th>

                <th field="level" width="20">LEVEL</th>

                <th field="hak_akses" width="70">HAK

AKSES</th>

            </tr>

        </thead>

    </table>

    <div id="toolbar">

```

```
        <a href="#" class="easyui-linkbutton" iconCls="icon-add"
plain="true" onclick="newData()">Tambah</a>
```

```
        <a href="#" class="easyui-linkbutton" iconCls="icon-edit"
plain="true" onclick="editData()">Edit</a>
```

```
        <a href="#" class="easyui-linkbutton" iconCls="icon-
remove" plain="true" onclick="removeData()">Hapus</a>
```

```
    </div>
```

```
    <div id="dlg" class="easyui-dialog"
style="width:400px;height:280px;padding:10px 20px"
```

```
        closed="true" buttons="#dlg-buttons">
```

```
        <div class="ftitle">Informasi User</div>
```

```
        <form id="fm" method="post" novalidate>
```

```
            <div class="fitem">
```

```
                <label>Id user:</label>
```

```
                <input name="id_user" id="id_user"
class="easyui-validatebox" required="true"/>
```

```
            </div>
```

```
            <div class="fitem">
```

```
                <label>Nama User:</label>
```

```
                <input name="nm_user" class="easyui-
validatebox" required="true">
```

```
            </div>
```

```
            <div class="fitem">
```

```
                <label>Hak Akses:</label>
```

```
                <select id="hak-akses" name="level"
style="width:120px;" required="true">
```

```

                                <option
value="admin">admin</option>

                                <option
value="gudang">gudang</option>

                                <option value="ppic">inventory
material</option>

                                </select>

                                </div>

                                <?php foreach($configs['hak_akses'] as $hak_akses)
:??>

                                <input type="checkbox" name="hak_akses[]"
value="<?php echo $hak_akses;?>" class='hak_akses' />

                                <?php echo $hak_akses;?>

                                <br/>

                                <?php endforeach;?>

                                <div class="fitem">

                                    <label>Password:</label>

                                    <input name="password" type="password"
class="easyui-validatebox" required="true">

                                </div>

                                </form>

                                <div id="dlg-buttons">

                                    <a href="#" class="easyui-linkbutton"
iconCls="icon-ok" onclick="saveData()">Save</a>

                                    <a href="#" class="easyui-linkbutton"
iconCls="icon-cancel" onclick="javascript:$('#dlg').dialog('close')">Cancel</a>

```

```
</div>

</body>

</html>
```

Akses

Login Gudang

```
<?php

session_start();

$is_ajax = $_REQUEST['is_ajax'];

if (isset($is_ajax) && $is_ajax) {

    $username = $_REQUEST['username'];

    $password = $_REQUEST['password'];

    include ('../koneksi/koneksi.php');

    $sql = mysql_query("SELECT * FROM user WHERE id_user='$username'
AND password='$password'");

    $num_rows = mysql_num_rows($sql);

    if ($num_rows > 0) {

        $rows = mysql_fetch_array($sql);

        $id_user = $rows['id_user'];

        $nm_user = $rows['nm_user'];

        $level = $rows['level'];

        $_SESSION['id_user'] = $id_user;
```

```

$_SESSION['nm_user'] = $nm_user;

$_SESSION['level'] = $level;

$_SESSION['hak_akses'] = explode('|', $rows['hak_akses']);

echo "success";

}

}

?>

```

Login Outlet

```

<?php

session_start();

$is_ajax = $_REQUEST['is_ajax'];

if(isset($is_ajax) && $is_ajax){

    $username = $_REQUEST['username'];

    $password = $_REQUEST['password'];

    include('../koneksi/koneksi.php');

    $sql=mysql_query("SELECT * FROM outlet WHERE
id_outlet='$username' AND password='$password'");

    $num_rows=mysql_num_rows($sql);

    if($num_rows>0){

        $rows=mysql_fetch_array($sql);

        $id_outlet=$rows['id_outlet'];

        $nm_outlet=$rows['nm_outlet'];

        $_SESSION['id_outlet']=$id_outlet;

```

```

        $_SESSION['nm_outlet']=$nm_outlet;

        echo "success";

    }

}

?>

```

Data Master Barang

get_barang.php

```

<?php

$page = isset($_POST['page']) ? intval($_POST['page']) : 1;

$rows = isset($_POST['rows']) ? intval($_POST['rows']) : 10;

$offset = ($page-1)*$rows;

$result = array();

include('../koneksi/koneksi.php');

$rs = mysql_query("SELECT COUNT(*) FROM barang");

$row = mysql_fetch_row($rs);

$result["total"] = $row[0];

$rs = mysql_query("

    SELECT * FROM barang, supplier

    WHERE barang.id_supplier = supplier.id_supplier

    LIMIT $offset,$rows

");

```

```
$items = array();

while($row = mysql_fetch_object($rs)){
    array_push($items, $row);
}

$result["rows"] = $items;

echo json_encode($result);

?>
```

Get id

```
<?php
```

```
include("../koneksi/koneksi.php");
```

```
$q_max_id = mysql_query("SELECT MAX(id_barang) as max_id FROM barang
LIMIT 1");
```

```
$data_max_id = mysql_fetch_assoc($q_max_id);
```

```
$curr_max_id = intval(substr($data_max_id['max_id'], -5));
```

```
$next_max_id = $curr_max_id+1;
```

```
$next_id = "B".str_pad($next_max_id, 7, '0', STR_PAD_LEFT);
```

```
echo json_encode(array('max_id' => $next_id));
```

Remove Barang

```
<?php
```

```
$id_barang = $_REQUEST['id_barang'];
```

```
include('../koneksi/koneksi.php');
```

```
$sql = "DELETE FROM barang WHERE id_barang='$id_barang'";
```

```
$result = @mysql_query($sql);
```

```
if ($result){
```

```
echo json_encode(array('success'=>true));
```

```
} else {
```

```
echo json_encode(array('msg'=>'Some errors occured.'));
```

```
}
```

```
?>
```

Save Barang

```
<?php
```

```
$id_barang = $_REQUEST['id_barang'];
```

```
$nm_barang = $_REQUEST['nm_barang'];
```

```
$stok = $_REQUEST['stok'];
```

```
$stok_min = $_REQUEST['stok_minimum'];
```

```
$satuan = $_REQUEST['satuan'];
```

```
$hrg_brg = $_REQUEST['hrg_brg'];
```

```

$hrng_smpn = $_REQUEST['hrng_smpn'];

$hrng_pesan = $_REQUEST['hrng_pesan'];

$permintaan = $_REQUEST['permintaan'];

$id_supplier = $_REQUEST['id_supplier'];


include ('../../koneksi/koneksi.php');


$sql = "

        INSERT INTO
barang(id_barang,nm_barang,stok,stok_minimum,satuan,hrng_brg,hrng_smpn,hrng_p
esan,permintaan,id_supplier)

        VALUES('$id_barang','$nm_barang','$stok','$stok_min','$satuan','$hrng_brg'
,'$hrng_smpn','$hrng_pesan','$permintaan', '$id_supplier')

";

$result = @mysql_query($sql);

if ($result) {

    echo json_encode(array('success' => true));

} else {

    echo json_encode(array('msg'=>'Maaf, ID tersebut sudah digunakan, coba
masukkan dengan ID yang lain.'));

    /* echo json_encode(array('msg' => mysql_error())); */

}

?>

```

Update Barang

```
<?php
```

```
$id_barang = $_REQUEST['id_barang'];
```

```
$nm_barang = $_REQUEST['nm_barang'];
```

```
$stok = $_REQUEST['stok'];
```

```
$stok_min = $_REQUEST['stok_minimum'];
```

```
$satuan = $_REQUEST['satuan'];
```

```
$hrg_brg = $_REQUEST['hrg_brg'];
```

```
$hrg_smpn = $_REQUEST['hrg_smpn'];
```

```
$hrg_pesan = $_REQUEST['hrg_pesan'];
```

```
$permintaan = $_REQUEST['permintaan'];
```

```
$id_supplier = $_REQUEST['id_supplier'];
```

```
include('../koneksi/koneksi.php');
```

```
$sql = "UPDATE barang SET id_supplier='$id_supplier',  
nm_barang='$nm_barang', stok='$stok', stok_minimum=$stok_min,  
satuan='$satuan', hrg_brg='$hrg_brg', hrg_smpn='$hrg_smpn',  
hrg_pesan='$hrg_pesan', permintaan='$permintaan' WHERE  
id_barang='$id_barang'";
```

```
$result = @mysql_query($sql);
```

```
if ($result){
```

```
echo json_encode(array('success'=>true));
```

```
} else {  
  
echo json_encode(array('msg'=>'Some errors occurred.));  
  
}  
  
>
```

Jenis Barang

Get_jenis

```
<?php  
  
$page = isset($_POST['page']) ? intval($_POST['page']) : 1;  
  
$rows = isset($_POST['rows']) ? intval($_POST['rows']) : 10;  
  
$offset = ($page-1)*$rows;  
  
$result = array();  
  
  
include('../koneksi/koneksi.php');  
  
  
$rs = mysql_query("SELECT COUNT(*) FROM jenis_barang");  
  
$row = mysql_fetch_row($rs);  
  
$result["total"] = $row[0];  
  
$rs = mysql_query("SELECT * FROM jenis_barang LIMIT  
$offset,$rows");  
  
  
$items = array();  
  
while($row = mysql_fetch_object($rs)){
```

```
        array_push($items, $row);
    }

    $result["rows"] = $items;

    echo json_encode($result);

?>
```

Remove Jenis

```
<?php

$id_jenis = $_REQUEST['id_jenis'];

include('../koneksi/koneksi.php');

$sql = "DELETE FROM jenis_barang WHERE id_jenis='$id_jenis'";
$result = @mysql_query($sql);

if ($result){

    echo json_encode(array('success'=>true));

} else {

    echo json_encode(array('msg'=>'Some errors occurred.'));

}

?>
```

Save Jenis

```
<?php

$id_jenis = $_REQUEST['id_jenis'];

$nm_jenis = $_REQUEST['nm_jenis'];


include('../koneksi/koneksi.php');


$sql = "INSERT INTO jenis_barang(id_jenis,nm_jenis)
VALUES('$id_jenis','$nm_jenis')";

$result = @mysql_query($sql);

if ($result){

    echo json_encode(array('success'=>true));

} else {

    echo json_encode(array('msg'=>'Some errors occurred.'));

}

?>
```

Update Jenis

```
<?php

$id_jenis = $_REQUEST['id_jenis'];

$nm_jenis = $_REQUEST['nm_jenis'];


include('../koneksi/koneksi.php');
```

```

$sql = "UPDATE jenis_barang SET nm_jenis='$nm_jenis' WHERE
id_jenis='$id_jenis'";

$result = @mysql_query($sql);

if ($result){

    echo json_encode(array('success'=>true));

} else {

    echo json_encode(array('msg'=>'Some errors occurred.'));

}

?>

```

Get Outlet

```

<?php

$page = isset($_POST['page']) ? intval($_POST['page']) : 1;

$rows = isset($_POST['rows']) ? intval($_POST['rows']) : 10;

$offset = ($page-1)*$rows;

$result = array();

include('../koneksi/koneksi.php');

$rs = mysql_query("SELECT COUNT(*) FROM outlet");

$row = mysql_fetch_row($rs);

$result["total"] = $row[0];

$rs = mysql_query("SELECT * FROM outlet LIMIT $offset,$rows");

```

```

        $items = array();

        while($row = mysql_fetch_object($rs)){

            array_push($items, $row);

        }

        $result["rows"] = $items;


        echo json_encode($result);

?>

```

Remove outlet

```

<?php

$id_outlet = $_REQUEST['id_outlet'];

include('../../koneksi/koneksi.php');

$sql = "DELETE FROM outlet WHERE id_outlet='$id_outlet'";

$result = @mysql_query($sql);

if ($result){

    echo json_encode(array('success'=>true));

} else {

    echo json_encode(array('msg'=>'Some errors occurred.'));

}

?>

```

Save Outlet

```
<?php

$id_outlet = $_REQUEST['id_outlet'];

$nm_outlet = $_REQUEST['nm_outlet'];

$almt = $_REQUEST['almt_outlet'];

$tlp = $_REQUEST['tlp_outlet'];

$pass = $_REQUEST['password'];


include('../koneksi/koneksi.php');


$sql = "INSERT INTO
outlet(id_outlet,nm_outlet,almt_outlet,tlp_outlet,password)
VALUES('$id_outlet','$nm_outlet','$almt','$tlp','$pass')";

$result = @mysql_query($sql);

if ($result){

    echo json_encode(array('success'=>true));

} else {

    echo json_encode(array('msg'=>'Some errors occured.'));

}

?>
```

Update Outlet

```
<?php

$id_outlet = $_REQUEST['id_outlet'];
```

```

$nm_outlet = $_REQUEST['nm_outlet'];

$almt = $_REQUEST['almt_outlet'];

$tlp = $_REQUEST['tlp_outlet'];

$pass = $_REQUEST['password'];


include('../../koneksi/koneksi.php');


$sql = "UPDATE outlet SET nm_outlet='$nm_outlet', almt_outlet='$almt',
tlp_outlet='$tlp', password='$pass' WHERE id_outlet='$id_outlet'";

$result = @mysql_query($sql);

if ($result){

    echo json_encode(array('success'=>true));

} else {

    echo json_encode(array('msg'=>'Some errors occurred.));

}

?>

```

Get Supplier

```

<?php

$page = isset($_POST['page']) ? intval($_POST['page']) : 1;

$rows = isset($_POST['rows']) ? intval($_POST['rows']) : 10;

$offset = ($page-1)*$rows;

$result = array();

```

```

include('../..//koneksi/koneksi.php');

$rs = mysql_query("SELECT COUNT(*) FROM supplier");
$row = mysql_fetch_row($rs);
$result["total"] = $row[0];

$rs = mysql_query("SELECT * FROM supplier LIMIT $offset,$rows");

$item = array();

while($row = mysql_fetch_object($rs)){
    array_push($item, $row);
}

$result["rows"] = $item;

echo json_encode($result);

?>

```

Remove Supplier

```

<?php

$id_supplier = $_REQUEST['id_supplier'];

include('../..//koneksi/koneksi.php');

$sql = "DELETE FROM supplier WHERE id_supplier='$id_supplier'";

$result = @mysql_query($sql);

```

```

if ($result){
    echo json_encode(array('success'=>true));
} else {
    echo json_encode(array('msg'=>'Some errors occurred.));
}
?>

```

Save Supplier

```

<?php

$id_supplier = $_REQUEST['id_supplier'];
$nm_supplier = $_REQUEST['nm_supplier'];
$almt = $_REQUEST['almt_supplier'];
$tlp = $_REQUEST['tlp_supplier'];

include('../koneksi/koneksi.php');

$sql = "INSERT INTO
supplier(id_supplier,nm_supplier,almt_supplier,tlp_supplier)
VALUES('$id_supplier','$nm_supplier','$almt','$tlp')";

$result = @mysql_query($sql);

if ($result){
    echo json_encode(array('success'=>true));
} else {

```

```
        echo json_encode(array('msg'=>'Maaf, ID tersebut sudah digunakan, coba
masukkan dengan ID yang lain.'));
    }
}
```

Update Supplier

```
<?php

$id_supplier = $_REQUEST['id_supplier'];
$nm_supplier = $_REQUEST['nm_supplier'];
$almt = $_REQUEST['almt_supplier'];
$tlp = $_REQUEST['tlp_supplier'];

include('../..//koneksi/koneksi.php');

$sql = "UPDATE supplier SET nm_supplier='$nm_supplier',
almt_supplier='$almt', tlp_supplier='$tlp' WHERE id_supplier='$id_supplier'";

$result = @mysql_query($sql);

if ($result){

    echo json_encode(array('success'=>true));

} else {

    echo json_encode(array('msg'=>'Some errors occured.'));

}

?>
```

Get User

```
<?php
```

```
$page = isset($_POST['page']) ? intval($_POST['page']) : 1;
```

```
$rows = isset($_POST['rows']) ? intval($_POST['rows']) : 10;
```

```
$offset = ($page-1)*$rows;
```

```
$result = array();
```

```
include('../koneksi/koneksi.php');
```

```
$rs = mysql_query("SELECT COUNT(*) FROM user");
```

```
$row = mysql_fetch_row($rs);
```

```
$result["total"] = $row[0];
```

```
$rs = mysql_query("SELECT * FROM user LIMIT $offset,$rows");
```

```
$items = array();
```

```
while($row = mysql_fetch_array($rs)){
```

```
    $row['hak_akses'] = implode(' ', explode('|', $row['hak_akses']));
```

```
    array_push($items, $row);
```

```
}
```

```
$result["rows"] = $items;
```

```
echo json_encode($result);
```

```
?>
```

Remove User

```
<?php

$id_user = $_REQUEST['id_user'];

include('../koneksi/koneksi.php');

$sql = "DELETE FROM user WHERE id_user='$id_user'";
$result = @mysql_query($sql);
if ($result){
    echo json_encode(array('success'=>true));
} else {
    echo json_encode(array('msg'=>'Some errors occurred.'));
}

?>
```

Save User

```
<?php

$id_user = $_REQUEST['id_user'];

$nm_user = $_REQUEST['nm_user'];

$pass = $_REQUEST['password'];

$hak_akses = $_REQUEST['hak_akses'];
```

```
$level = $_REQUEST['level'];
```

```
include('../../koneksi/koneksi.php');
```

```
$hak_akses = implode('|', $hak_akses);
```

```
$sql = "
```

```
    INSERT INTO user
```

```
    (id_user,nm_user,password,hak_akses,level)
```

```
    VALUES
```

```
    ('$id_user','$nm_user','$pass','$hak_akses','$level')
```

```
";
```

```
$result = @mysql_query($sql);
```

```
if ($result){
```

```
    echo json_encode(array('success'=>true));
```

```
} else {
```

```
    echo json_encode(array('msg'=>'Maaf, ID tersebut sudah digunakan, coba  
    masukkan dengan ID yang lain.'));
```

```
}
```

```
?>
```

Update User

```
<?php

$id_user = $_REQUEST['id_user'];

$nm_user = $_REQUEST['nm_user'];

$pass = $_REQUEST['password'];

//$level = $_REQUEST['level'];

$hak_akses = implode('|', (array) $_REQUEST['hak_akses']);

include('../../koneksi/koneksi.php');

$sql = "UPDATE user SET
nm_user='$nm_user',password='$pass',hak_akses='$hak_akses' WHERE
id_user='$id_user'";

$result = @mysql_query($sql);

if ($result){

    echo json_encode(array('success'=>true));

} else {

    echo json_encode(array('msg'=>'Some errors occured.'));

}
```

